

Vassil N. Alexandrov
Geert Dick van Albada
Peter M.A. Sloot
Jack Dongarra (Eds.)

LNCS 3993

Computational Science – ICCS 2006

6th International Conference
Reading, UK, May 2006
Proceedings, Part III

3
Part III

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Vassil N. Alexandrov
Geert Dick van Albada Peter M.A. Sloot
Jack Dongarra (Eds.)

Computational Science – ICCS 2006

6th International Conference
Reading, UK, May 28-31, 2006
Proceedings, Part III

Volume Editors

Vassil N. Alexandrov
University of Reading
Centre for Advanced Computing and Emerging Technologies
Reading RG6 6AY, UK
E-mail: v.n.alexandrov@rdg.ac.uk

Geert Dick van Albada
Peter M.A. Sloot
University of Amsterdam
Department of Mathematics and Computer Science
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: {dick,sloot}@science.uva.nl

Jack Dongarra
University of Tennessee
Computer Science Department
1122 Volunteer Blvd., Knoxville, TN 37996-3450, USA
E-mail: dongarra@cs.utk.edu

Library of Congress Control Number: 2006926429

CR Subject Classification (1998): F, D, G, H, I, J, C.2-3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-34383-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34383-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11758532 06/3142 5 4 3 2 1 0

Preface

The Sixth International Conference on Computational Science (ICCS 2006) was held in Reading, United Kingdom, May 28-31 and continued the traditions of previous conferences in the series: ICCS 2005 in Atlanta, Georgia, USA; ICCS 2004 in Krakow, Poland; ICCS 2003 held simultaneously at two locations in, Melbourne, Australia and St. Petersburg, Russia; ICCS 2002 in Amsterdam, The Netherlands; and ICCS 2001 in San Francisco, California, USA.

Since the first conference in San Francisco, rapid developments in Computational Science as a mainstream area facilitating multi-disciplinary research essential for the advancement of science have been observed. The theme of ICCS 2006 was “Advancing Science through Computation”, marking several decades of progress in Computational Science theory and practice, leading to greatly improved applications science. The conference focused on the following major themes: tackling Grand Challenges Problems; modelling and simulations of complex systems; scalable algorithms and tools and environments for Computational Science. Of particular interest were the following major recent developments in novel methods and modelling of complex systems for diverse areas of science, scalable scientific algorithms, advanced software tools, computational grids, advanced numerical methods, and novel application areas where the above novel models, algorithms and tools can be efficiently applied such as physical systems, computational and systems biology, environmental systems, finance, and others.

Keynote lectures were delivered by Mateo Valero (Director, Barcelona Supercomputing Centre) - “Tackling Grand Challenges Problems”; Chris Johnson (Distinguished Professor, University of Utah) - “Visualizing the Future”; José Moreira (IBM, Chief Architect, Commercial Scale Out) - “Achieving Breakthrough Science with the Blue Gene/L Supercomputer”; Martin Curley (INTEL, Global Director of Innovation and IT Research) - “IT Innovation: A New Era”; Vaidy Sunderam (Samuel Candler Dobbs Professor of Computer Science, Emory University, USA) - “Metacomputing Revisited: Alternative Paradigms for Distributed Resource Sharing”; and Ron Bell (AWE plc.) - “The AWE HPC Benchmark”.

In addition, two special sessions were held - one by industry and one by the funding bodies. Three tutorials preceded the main technical program of the conference: “Tools for Program Analysis in Computational Science” by Dieter Kranzlmüller; “P-GRADE Portal” by P. Kascuk, T. Kiss and G. Sipos; and “Scientific Computing on Graphics Hardware” by Dominik Göddeke. We would like to thank all the keynote, the invited, and the tutorial speakers for their inspiring talks.

Apart from the plenary sessions and tutorials the conference included twelve parallel oral sessions and two poster sessions. Since the first ICCS in San

Francisco the conference has grown steadily attracting increasing numbers of researchers in the field of Computational Science. For ICCS 2006 we received over 1,400 submissions, around 300 for the main track and over 1,100 for the originally proposed workshops. Of these submissions, 98 were accepted as full papers and 29 as posters for the main track; and 500 were accepted as full papers, short papers or posters for the 32 workshops. This selection was possible due to the tremendous work done by the Program Committee and the 720 reviewers. The author index contains over 1,000 names and over 600 participants from all the major continents. The papers cover a wide variety of topics in Computational Science, ranging from Grand Challenges problems and modelling of complex systems in various areas to advanced numerical algorithms and new scalable algorithms in diverse application areas and software environments for Computational Science. The ICCS 2006 Proceedings consist of four volumes, 3991 to 3994, where the first volume contains the papers from the main track and all the posters; the remaining three volumes contain the papers from the workshops. ICCS this year is primarily published on a CD and we would like to thank Springer for their cooperation and partnership. We hope that the ICCS 2006 Proceedings will be a major intellectual resource for many computational scientists and researchers for years ahead. During the conference the best papers from the main track and workshops as well as the best posters were nominated and commended on ICCS 2006 website. A number of selected papers will also be published in special issues of relevant mainstream journals.

We would like to thank all workshop organisers and the program committee for the excellent work, which further enhanced the conference's standing and led to very high quality event with excellent papers. We would like to express our gratitude to Advanced Computing and Emerging Technologies Centre staff, postgraduates and students for their wholehearted support of ICCS 2006. We would like to thank the School of Systems Engineering, Conference Office, Finance Department and various units at the University of Reading for different aspects of the organization and for their constant support in making ICCS 2006 a success. We would like to thank the Local Organizing Committee for their persistent and enthusiastic work towards the success of ICCS 2006. We owe special thanks to our sponsors: Intel, IBM, SGI, Microsoft Research, EPSRC and Springer; and to ACET Centre and the University of Reading for their generous support. We would like to thank SIAM, IMACS, and UK e-Science programme for endorsing ICCS 2006.

ICCS 2006 was organized by the Advanced Computing and Emerging Technologies Centre, University of Reading, with support from the Section Computational Science at the Universiteit van Amsterdam and Innovative Computing Laboratory at the University of Tennessee, in cooperation with the Society for Industrial and Applied Mathematics (SIAM), the International Association for Mathematics and Computers in Simulation (IMACS), and the UK Engineering and Physical Sciences Research Council (EPSRC). We invite you to visit the ICCS 2006 website (<http://www.iccs-meeting.org/iccs2006/>) and ACET Centre website (<http://www.acet.reading.ac.uk/>) to recount the events leading up

to the conference, to view the technical programme, and to recall memories of three and a half days of engagement in the interest of fostering and advancing Computational Science.

June 2006

Vassil N. Alexandrov
G. Dick van Albada
Peter M.A. Sloot
Jack J. Dongarra

Organisation

ICCS 2006 was organised by the Centre for Advanced Computing and Emerging Technologies (ACET), University of Reading, UK, in cooperation with the University of Reading (UK), the Universiteit van Amsterdam (The Netherlands), the University of Tennessee (USA), Society for Industrial and Applied Mathematics (SIAM), International Association for Mathematics and Computers in Simulation (IMACS) and Engineering and Physical Sciences Research Council (EPSRC). The conference took place on the Whiteknights Campus of the University of Reading.

Conference Chairs

Scientific Chair - Vassil N. Alexandrov (ACET, University of Reading, UK)

Workshops Chair - G. Dick van Albada (Universiteit van Amsterdam, The Netherlands)

ICCS Series Overall Chair - Peter M.A. Sloot (Universiteit van Amsterdam, The Netherlands)

ICCS Series Overall Co-Chair - Jack J. Dongarra (University of Tennessee, USA)

Local Organising Committee

Vassil N. Alexandrov

Linda Mogort-Valls

Nia Alexandrov

Ashish Thandavan

Christian Weihrauch

Simon Branford

Adrian Haffegge

David Monk

Janki Dodiya

Priscilla Ramsamy

Ronan Jamieson

Ali Al-Khalifah

David Johnson

Eve-Marie Larsen

Gareth Lewis

Ismail Bhana

S. Mehmood Hasan

Sokratis Antoniou

Sponsoring Institutions

Intel Corporation
IBM
SGI
Microsoft Research
EPSRC
Springer
ACET Centre
University of Reading

Endorsed by

SIAM
IMACS
UK e-Science Programme

Program Committee

D. Abramson - Monash University, Australia
V. Alexandrov - University of Reading, UK
D.A. Bader - Georgia Tech, USA
M. Baker - University of Portsmouth, UK
S. Belkasim - Georgia State University, USA
A. Benoit - Ecole Normale Supérieure de Lyon, France
I. Bhana - University of Reading, UK
R. Blais - University of Calgary, Canada
A. Bogdanov - Institute for High Performance Computing and Information Systems, Russia
G. Bosilca - University of Tennessee, USA
S. Branford - University of Reading, UK
M. Bubak - Institute of Computer Science and ACC Cyfronet - AGH, Poland
R. Buyya - University of Melbourne, Australia
F. Cappello - Laboratoire de Recherche en Informatique, Paris Sud, France
T. Cortes - Universitat Politècnica de Catalunya, Spain
J.C. Cunha - New University of Lisbon, Portugal
F. Desprez - INRIA, France
T. Dhaene - University of Antwerp, Belgium
I.T. Dimov - University of Reading, UK
J. Dongarra - University of Tennessee, USA
C. Douglas - University of Kentucky, USA
G.E. Fagg, University of Tennessee, USA
M. Gerndt - Technical University of Munich, Germany

- Y. Gorbachev - Institute for High Performance Computing and Information Systems, Russia
- A. Goscinski - Deakin University, Australia
- A. Haffegge - University of Reading, UK
- L. Hluchy - Slovak Academy of Science, Slovakia
- A. Hoekstra - Universiteit van Amsterdam, The Netherlands
- A. Iglesias - University of Cantabria, Spain
- R. Jamieson - University of Reading, UK
- D. Johnson - University of Reading, UK
- J. Kitowski - AGH University of Science and Technology, Poland
- D. Kranzlmüller - Johannes Kepler University Linz, Austria
- A. Lagana - Università di Perugia, Italy
- G. Lewis - University of Reading, UK
- E. Luque - University Autònoma of Barcelona, Spain
- M. Malawski - Institute of Computer Science AGH, Poland
- M. Mascagni - Florida State University, USA
- E. Moreno - Euripides Foundation of Marilia, Brazil
- J. Ni The - University of Iowa, Iowa City, IA, USA
- G. Norman - Russian Academy of Sciences, Russia
- S. Orlando - University of Venice, Italy
- B. Ó Nulláin - UUniversiteit van Amsterdam, The Netherlands
- M. Paprzycki - Computer Science Institute, SWSP, Warsaw, Poland
- R. Perrott - Queen's University of Belfast, UK
- R. Renaut - Arizona State University, USA
- A. Rendell - Australian National University, Australia
- D. Rodriguez-García - University of Reading, UK
- P. Roe Queensland - University of Technology, Australia
- S.L. Scott - Oak Ridge National Laboratory, USA
- D. Shires - U.S. Army Research Laboratory, USA
- P.M.A. Sloot - Universiteit van Amsterdam, The Netherlands
- G. Stuer - University of Antwerp, Belgium
- R. Tadeusiewicz - AGH University of Science and Technology, Poland
- A. Thandavan - University of Reading, UK
- P. Tvrdik - Czech Technical University, Czech Republic
- P. Uthayopas - Kasetsart University, Thailand
- G.D. van Albada - Universiteit van Amsterdam, The Netherlands
- J. Vigo-Aguiar - University of Salamanca, Spain
- J.A. Vrugt - Los Alamos National Laboratory, USA
- J. Wasniewski - Technical University of Denmark, Denmark
- G. Watson - Los Alamos National Laboratory, USA
- C. Weihrauch - University of Reading, UK
- Y. Xue - Chinese Academy of Sciences, China
- E. Zudilova-Seinstra - Universiteit van Amsterdam, The Netherlands

Reviewers

- | | | |
|-------------------|-------------------|-------------------|
| A. Adamatzky | A. Pieczynska | B. Shan |
| A. Arenas | A. Rackauskas | B. Sniezynski |
| A. Belloum | A. Rendell | B. Song |
| A. Benoit | A. Sánchez | B. Strug |
| A. Bielecki | A. Sánchez-Campos | B. Tadic |
| A. Bode | A. Sayyed-Ahmad | B. Xiao |
| A. Cepulkauskas | A. Shafarenko | B.M. Rode |
| A. Chkrebti | A. Skowron | B.S. Shin |
| A. Drummond | A. Sosnov | C. Anthes |
| A. Erzan | A. Sourin | C. Bannert |
| A. Fedaravicius | A. Stuempel | C. Biely |
| A. Galvez | A. Thandavan | C. Bischof |
| A. Gerbessiotis | A. Tiskin | C. Cotta |
| A. Goscinski | A. Turan | C. Douglas |
| A. Griewank | A. Walther | C. Faure |
| A. Grösslinger | A. Wei | C. Glasner |
| A. Grzech | A. Wibisono | C. Grelck |
| A. Haffeege | A. Wong | C. Herrmann |
| A. Hoekstra | A. Yacizi | C. Imielinska |
| A. Iglesias | A. Zelikovsky | C. Lursinsap |
| A. Jakulin | A. Zhmakin | C. Mastroianni |
| A. Janicki | A. Zhou | C. Miyaji |
| A. Javor | A.N. Karaivanova | C. Nelson |
| A. Karpfen | A.S. Rodinov | C. Otero |
| A. Kertész | A.S. Tosun | C. Rodriguez Leon |
| A. Knuepfer | A.V. Bogdanov | C. Schaubschläger |
| A. Koukam | B. Ó Nualláin | C. Wang |
| A. Lagana | B. Autin | C. Weihrauch |
| A. Lawniczak | B. Balis | C. Woolley |
| A. Lewis | B. Boghosian | C. Wu |
| A. Li | B. Chopard | C. Xu |
| A. Ligeza | B. Christianson | C. Yang |
| A. Mamat | B. Cogan | C.-H. Huang |
| A. Martin del Rey | B. Dasgupta | C.-S. Jeong |
| A. McGough | B. Di Martino | C.G.H. Diks |
| A. Menezes | B. Gabrys | C.H. Goya |
| A. Motter | B. Javadi | C.H. Kim |
| A. Nasri | B. Kahng | C.H. Wu |
| A. Neumann | B. Kovalerchuk | C.K. Chen |
| A. Noel | B. Lesyng | C.N. Lee |
| A. Obuchowicz | B. Paternoster | C.R. Kleijn |
| A. Papini | B. Payne | C.S. Hong |
| A. Paventhan | B. Saunders | D. Abramson |

D. Brinza	E. Nawarecki	G. Mauri
D. Brown	E. Puppo	G. Messina
D. Che	E. Roanes-Lozano	G. Mounié
D. Déry	E. Valakevicius	G. Narasimhan
D. Donnelly	E. Zeng	G. Norman
D. Evers	E. Zotenko	G. Pavesi
D. Göddeke	E. Zudilova-Seinstra	G. Rojek
D. Johnson	E.A. Castro	G. Slusarczyk
D. Kim	E.N. Huh	G. Stuer
D. Kranzlmüller	E.S. Quintana-Orti	G. Szabó
D. Laforenza	F. Capkovic	G. Tempesti
D. Li	F. Cappello	G. Volkert
D. Luebke	F. Desprez	G. Watson
D. Maringer	F. Gava	G. Zheng
D. Pfahl	F. Hirata	G.-L. Park
D. Plemenos	F. Iavernaro	G.D. van Albada
D. Rodriguez-García	F. Kiss	G.D. Vedova
D. Shires	F. Lamantia	G.E. Fagg
D. Stoffer	F. Lee	G.J. Rodgers
D. Stokic	F. Loulergue	H. Bungartz
D. Szczerba	F. Markowetz	H. Choo
D. Taniar	F. Melendez	H. Diab
D. Thalmann	F. Perales	H. Fangohr
D. Vasuinin	F. Rogier	H. Jin
D. Wang	F. Terpstra	H. Kaltenbach
D. Xu	F. Zuccarello	H. Kosina
D.A. Bader	F.-X. Roux	H. Labiod
D.B. Davies	F.J. Keil	H. Lee
D.B.D. Birkbeck	G. Alexe	H. Moradkhani
D.C. Ghosh	G. Allen	H. Müller
D.C. Lee	G. Bosilca	H. Munakata
D.J. Roberts	G. Chen	H. Oh
D.M. Chiu	G. Cheng	H. Sarafian
D.M. Tartakovsky	G. Dobrowolski	H. Stockinger
D.R. Green	G. Dong	H. Suzuki
D.S. Kim	G. Erlebacher	H. Umeo
D.S. Perry	G. Farin	H. Wang
E. Atanasov	G. Felici	H. Yanami
E. Grabska	G. Frenking	H.-K. Choi
E. Huedo Cuesta	G. Gheri	H.-K. Lee
E. Jaeger-Frank	G. Jeon	H.C. Chojnacki
E. Lee	G. Kolaczek	H.F. Schaefer III
E. Luque	G. Kou	H.K. Kim
E. Macias	G. Lewis	H.P. Luehi
E. Moreno	G. Lin	H.S. Nguyen

H.Y. Lee	J. Kroc	J.J. Korczak
I. Bhana	J. Krueger	J.J. Zhang
I. Boada	J. Laws	J.K. Choi
I. Kolingerova	J. Lee	J.L. Leszczynski
I. Lee	J. Li	J.M. Bradshaw
I. Mandoiu	J. Liu	J.M. Gilp
I. Moret	J. Michopoulos	J.P. Crutchfield
I. Navas-Delgado	J. Nabrzyski	J.P. Suarez Rivero
I. Podolak	J. Nenortaite	J.V. Alvarez
I. Schagaev	J. Ni	J.Y. Chen
I. Suehiro	J. Owen	K. Akkaya
I. Tabakow	J. Owens	K. Anjyo
I. Taylor	J. Pang	K. Banas
I.T. Dimov	J. Pjesivac-Grbovic	K. Bolton
J. Abawajy	J. Quinqueton	K. Boryczko
J. Aroba	J. Sanchez-Reyes	K. Chae
J. Blower	J. Shin	K. Ebihara
J. Cabero	J. Stefanowski	K. Ellrott
J. Cai	J. Stoye	K. Fisher
J. Cao	J. Tao	K. Fuerlinger
J. Chen	J. Utke	K. Gaaloul
J. Cho	J. Vigo-Aguiar	K. Han
J. Choi	J. Volkert	K. Hsu
J. Davila	J. Wang	K. Jinsuk
J. Dolado	J. Wasniewski	K. Juszczyszyn
J. Dongarra	J. Weidendorfer	K. Kubota
J. Guo	J. Wu	K. Li
J. Gutierrez	J. Yu	K. Meridg
J. Han	J. Zara	K. Najarian
J. He	J. Zhang	K. Ouazzane
J. Heo	J. Zhao	K. Sarac
J. Hong	J. Zivkovic	K. Sycara
J. Humble	J.-H. Nam	K. Tai-hoon Kim
J. Hwang	J.-L. Koning	K. Trojahner
J. Jeong	J.-W. Lee	K. Tuncay
J. Jurek	J.A. Vrugt	K. Westbrook
J. Kalcher	J.C. Cunha	K. Xu
J. Kang	J.C. Liu	K. Yang
J. Kim	J.C. Teixeira	K. Zhang
J. King	J.C.S. Lui	K.-J. Jeong
J. Kitowski	J.F. San Juan	K.B. Lipkowitz
J. Koller	J.H. Hrusak	K.D. Nguyen
J. Kommineni	J.H. Lee	K.V. Mikkelsen
J. Koo	J.J. Alvarez	K.X.S. Souza
J. Kozlak	J.J. Cuadrado	K.Y. Huang

L. Borzemski	M. Hobbs	N. Sundaraganesan
L. Brugnano	M. Houston	N.T. Nguyen
L. Cai	M. Iwami	O. Beckmann
L. Czekierda	M. Jankowski	O. Belmonte
L. Fernandez	M. Khater	O. Habala
L. Gao	M. Kim	O. Maruyama
L. Gonzalez-Vega	M. Kirby	O. Otto
L. Hascoet	M. Kisiel-Dorochinicki	O. Yasar
L. Hluchy	M. Li	P. Alper
L. Jia	M. Malawski	P. Amodio
L. Kotulski	M. Mascagni	P. Balbuena
L. Liu	M. Morshed	P. Bekaert
L. Lopez	M. Mou	P. Berman
L. Marchal	M. Omar	P. Blowers
L. Neumann	M. Pérez-Hernández	P. Bonizzoni
L. Parida	M. Palakal	P. Buendia
L. Taher	M. Paprzycki	P. Czarnul
L. Xiao	M. Paszynski	P. Damaschke
L. Xin	M. Polak	P. Diaz Gutierrez
L. Yang	M. Rajkovic	P. Dyshlovenko
L. Yu	M. Ronsse	P. Geerlings
L. Zheng	M. Rosvall	P. Gruer
L. Zhigilei	M. Ruiz	P. Heimbach
L.H. Figueiredo	M. Sarfraz	P. Heinzlreiter
L.J. Song	M. Sbert	P. Herrero
L.T. Yang	M. Smolka	P. Hovland
M. Aldinucci	M. Suvakov	P. Kacsuk
M. Baker	M. Tomassini	P. Li
M. Bamha	M. Verleysen	P. Lingras
M. Baumgartner	M. Vianello	P. Martineau
M. Bhuruth	M. Zhang	P. Pan
M. Borodovsky	M.A. Sicilia	P. Praxmarer
M. Bubak	M.H. Zhu	P. Rice
M. Caliari	M.J. Brunger	P. Roe
M. Chover	M.J. Harris	P. Slood
M. Classen	M.Y. Chung	P. Tvrdik
M. Comin	N. Bauernfeind	P. Uthayopas
M. Deris	N. Hu	P. van Hooft
M. Drew	N. Ishizawa	P. Venuvanalingam
M. Fagan	N. Jayaram	P. Whitlock
M. Fras	N. Masayuki	P. Wolschann
M. Fujimoto	N. Murray	P.H. Lin
M. Gerndt	N. Navarro	P.K. Chattaraj
M. Guo	N. Navet	P.R. Ramasami
M. Hardman	N. Sastry	Q. Deng

R. Aspin	S. Dong	T. Ida
R. Blais	S. El Yacoubi	T. Korkmaz
R. Buyya	S. Forth	T. McKenzie
R. Dondi	S. Gilmore	T. Milledge
R. Drezewski	S. Gimelshein	T. Politi
R. Duran Diaz	S. Gorlatch	T. Przytycka
R. Jamieson	S. Green	T. Recio
R. Jothi	S. Gremalschi	T. Strothotte
R. Kakkar	S. Han	T. Suzudo
R. Katarzyniak	S. Jhang	T. Takahashi
R. Kobler	S. Kawano	T. Tsuji
R. Lambiotte	S. Kim	T. Wang
R. Liu	S. Lee	T. Ward
R. Marcjan	S. Lightstone	T. Worsch
R. Mikusauskas	S. Maniccam	T.-J. Lee
R. Nock	S. Olariu	T.B. Ho
R. Perrott	S. Orlando	T.C. Lu
R. Ramarosan	S. Pal	T.L. Zhang
R. Rejas	S. Rahmann	T.N. Troung
R. Renaut	S. Rajasekaran	T.V. Gurov
R. Rizzi	S. Sanchez	T.W. Kim
R. Ruiz	S. Thurner	U. Ruede
R. Sander	S. Tsunekawa	U. Ufuktepe
R. Schaefer	S. Turek	U. Vaccaro
R. Simutis	S. Valverde	U.N. Naumann
R. Strzodka	S. Yi	V. Alexandrov
R. Tadeusiewicz	S. Yoon	V. Aquilanti
R. Walentynski	S.-B. Scholz	V. Debelov
R. Westermann	S.-R. Kim	V. Hargy
R. Wismüller	S.-Y. Han	V. Korkhov
R. Wolff	S.C. Lo	V. Parasuk
R.G. Giering	S.H. Cho	V. Rafe
R.Q. Wu	S.J. Han	V. Robles
S. Abe	S.K. Ghosh	V. Srovnal
S. Aluru	S.L. Gargh	V. Weispfenning
S. Ambroszkiewicz	S.L. Scott	V.A. Emanuele II
S. Balla	S.S. Manna	V.C. Chinh
S. Bandini	T. Angskun	V.V. Krzhizhanovskaya
S. Belkasim	T. Atoguchi	V.V. Shakhov
S. Bhowmick	T. Cortes	W. Alda
S. Böcker	T. Dhaene	W. Bronsvort
S. Branford	T. Dokken	W. Choi
S. Chen	T. Ezaki	W. Dou
S. Chiu	T. Fahringer	W. Funika
S. Cho	T. Hu	W. Lee

W. Miller	Y. Cotronis	Y.J. Ye
W. Rachowicz	Y. Cui	Y.Q. Xiong
W. Yan	Y. Dai	Y.S. Choi
W. Yin	Y. Li	Y.Y. Cho
W. Zhang	Y. Liu	Y.Z. Cho
W. Zheng	Y. Mun	Z. Cai
W.K. Tai	Y. Pan	Z. Hu
X. Huang	Y. Peng	Z. Huang
X. Liao	Y. Shi	Z. Liu
X. Wan	Y. Song	Z. Pan
X. Wang	Y. Xia	Z. Toroczka
X. Zhang	Y. Xue	Z. Wu
X.J. Chen	Y. Young Jin	Z. Xin
X.Z. Cheng	Y.-C. Bang	Z. Zhao
Y. Aumann	Y.-C. Shim	Z. Zlatev
Y. Byun	Y.B. Kim	Z.G. Sun
Y. Cai	Y.E. Gorbachev	Z.M. Zhou

Workshop Organisers

Third International Workshop on Simulation of Multiphysics Multiscale Systems

V.V. Krzhizhanovskaya - Universiteit van Amsterdam, The Netherlands and
 St. Petersburg State Polytechnical University, Russia
 Y.E. Gorbachev - St. Petersburg State Polytechnic University, Russia
 B. Chopard - University of Geneva, Switzerland

Innovations in Computational Science Education

D. Donnelly - Department of Physics, Siena College, USA

Fifth International Workshop on Computer Graphics and Geometric Modeling (CGGM 2006)

A. Iglesias - University of Cantabria, Spain

Fourth International Workshop on Computer Algebra Systems and Applications (CASA 2006)

A. Iglesias - University of Cantabria, Spain
 A. Galvez - University of Cantabria, Spain

Tools for Program Development and Analysis in Computational Science

D. Kranzlmüller - GUP, Joh. Kepler University, Linz, Austria
R. Wismüller - University of Siegen, Germany
A. Bode - Technische Universität München, Germany
J. Volkert - GUP, Joh. Kepler University, Linz, Austria

Collaborative and Cooperative Environments

C. Anthes - GUP, Joh. Kepler University, Linz, Austria
V.N. Alexandrov - ACET, University of Reading, UK
D.J. Roberts - NICVE, University of Salford, UK
J. Volkert - GUP, Joh. Kepler University, Linz, Austria
D. Kranzlmüller - GUP, Joh. Kepler University, Linz, Austria

Second International Workshop on Bioinformatics Research and Applications (IWBRA'06)

A. Zelikovsky - Georgia State University, USA
Y. Pan - Georgia State University, USA
I.I. Mandoiu - University of Connecticut, USA

Third International Workshop on Practical Aspects of High-Level Parallel Programming (PAPP 2006)

A. Benoît - Laboratoire d'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, France
F. Loulergue - LIFO, Université d'Orléans, France

Wireless and Mobile Systems

H. Choo - Networking Laboratory, Sungkyunkwan University, Suwon, KOREA

GeoComputation

Y. Xue - Department of Computing, Communications Technology and Mathematics, London Metropolitan University, UK

Computational Chemistry and Its Applications

P. Ramasami - Department of Chemistry, University of Mauritius

Knowledge and Information Management in Computer Communication Systems (KIMCCS 2006)

N.T. Nguyen - Institute of Control and Systems Engineering, Wroclaw University of Technology, Poland

- A. Grzech - Institute of Information Science and Engineering,
Wroclaw University of Technology, Poland
- R. Katarzyniak - Institute of Information Science and Engineering,
Wroclaw University of Technology, Poland

Modelling of Complex Systems by Cellular Automata (MCSCA 2006)

- J. Kroc - University of West Bohemia, Czech Republic
T. Suzudo - Japan Atomic Energy Agency, Japan
S. Bandini - University of Milano - Bicocca, Italy

Dynamic Data Driven Application Systems (DDDAS 2006)

- F. Darema - National Science Foundation, USA

Parallel Monte Carlo Algorithms for Diverse Applications in a Distributed Setting

- I.T. Dimov - ACET, University of Reading, UK
V.N. Alexandrov - ACET, University of Reading, UK

International Workshop on Intelligent Storage Technology (IST06)

- J. Shu - Department of Computer Science and Technology, Tsinghua University,
Beijing, P.R. China

Intelligent Agents in Computing Systems

- R. Schaefer - Department of Computer Science, Stanislaw Staszic University
of Science and Technology in Kraków
K. Cetnarowicz - Department of Computer Science, Stanislaw Staszic University
of Science and Technology in Kraków

First International Workshop on Workflow Systems in e-Science (WSES06)

- Z. Zhao - Informatics Institute, University of Amsterdam, The Netherlands
A. Belloum - University of Amsterdam, The Netherlands

Networks: Structure and Dynamics

- B. Tadic - Theoretical Physics Department, J. Stefan Institute, Ljubljana,
Slovenia
S. Thurner - Complex Systems Research Group, Medical University Vienna,
Austria

Evolution Toward Next Generation Internet (ENGI)

Y. Cui - Tsinghua University, P.R. China

T. Korkmaz - University of Texas at San Antonio, USA

General Purpose Computation on Graphics Hardware (GPGPU): Methods, Algorithms and Applications

D. Göldeke - Universität Dortmund, Institut für Angewandte Mathematik
und Numerik, Germany

S. Turek - Universität Dortmund, Institut für Angewandte Mathematik
und Numerik, Germany

Intelligent and Collaborative System Integration Technology (ICSIT)

J.-W. Lee - Center for Advanced e-System Integration Technology,
Konkuk University, Seoul, Korea

Computational Methods for Financial Markets

R. Simutis - Department of Informatics, Kaunas Faculty, Vilnius University,
Lithuania

V. Sakalauskas - Department of Informatics, Kaunas Faculty, Vilnius University,
Lithuania

D. Kriksčiuniene - Department of Informatics, Kaunas Faculty,
Vilnius University, Lithuania

2006 International Workshop on P2P for High Performance Computational Sciences (P2P-HPCS06)

H. Jin - School of Computer Science and Technology, Huazhong University of
Science and Technology, Wuhan, China

X. Liao - Huazhong University of Science and Technology, Wuhan, China

Computational Finance and Business Intelligence

Y. Shi - Graduate School of the Chinese Academy of Sciences, Beijing, China

Third International Workshop on Automatic Differentiation Tools and Applications

C. Bischof - Inst. for Scientific Computing, RWTH Aachen University, Germany

S.A. Forth - Engineering Systems Department, Cranfield University,
RMCS Shrivenham, UK

U. Naumann - Software and Tools for Computational Engineering,
RWTH Aachen University, Germany

J. Utke - Mathematics and Computer Science Division, Argonne National
Laboratory, IL, USA

2006 Workshop on Scientific Computing in Electronics Engineering

Y. Li - National Chiao Tung University, Hsinchu City, Taiwan

New Trends in the Numerical Solution of Structured Systems with Applications

T. Politi - Dipartimento di Matematica, Politecnico di Bari, Itali

L. Lopez - Dipartimento di Matematica, Università di Bari, Itali

Workshop on Computational Science in Software Engineering (CSSE'06)

D. Rodríguez García - University of Reading, UK

J.J. Cuadrado - University of Alcalá, Spain

M.A. Sicilia - University of Alcalá, Spain

M. Ruiz - University of Cádiz, Spain

Digital Human Modeling (DHM-06)

Y. Cai - Carnegie Mellon University, USA

C. Imielinska - Columbia University

Real Time Systems and Adaptive Applications (RTSAA 06)

T. Kuo - National Taiwan University, Taiwan

J. Hong - School of Computer Science and Engineering, Kwangwoon University, Seoul, Korea

G. Jeon - Korea Polytechnic University, Korea

International Workshop on Grid Computing Security and Resource Management (GSRM'06)

J.H. Abawajy - School of Information Technology, Deakin University, Geelong, Australia

Fourth International Workshop on Autonomic Distributed Data and Storage Systems Management Workshop (ADSM 2006)

J.H. Abawajy - School of Information Technology, Deakin University, Geelong, Australia

Table of Contents – Part III

GeoComputation

Information Registry of Remotely Sensed Meta-module in Grid Environment <i>Yong Xue, Jianqin Wang, Chaolin Wu, Yincui Hu, Jianping Guo, Lei Zheng, Wei Wan, Guoyin Cai, Ying Luo, Shaobo Zhong</i>	1
Preliminary Study of Avian Influenza A Infection Using Remote Sensing and GIS Techniques <i>Jianping Guo, Yong Xue, Shaobo Zhong, Chunxiang Cao, Wuchun Cao, Xiaowen Li, Liqun Fang</i>	9
Efficient Coding of Quadtree Nodes <i>Mariano Pérez, Xaro Benavent, R. Olanda</i>	13
Special Task Scheduling and Control of Cluster Parallel Computing for High-Performance Ground Processing System <i>Wanjun Zhang, Dingsheng Liu, Guoqing Li, Wenyi Zhang</i>	17
AMEEPAR: Parallel Morphological Algorithm for Hyperspectral Image Classification on Heterogeneous Networks of Workstations <i>Antonio Plaza, Javier Plaza, David Valencia</i>	24
Visual Discovery and Reconstruction of the Climatic Conditions of the Past <i>Roberto Therón</i>	32
Per-pixel Rendering of Terrain Data <i>Taek Sang Jeong, JungHyun Han</i>	40
Spherical Harmonic Transforms Using Quadratures and Least Squares <i>J.A.R. Blais, M.A. Soofi</i>	48
Numerical Simulations of Space-Time Conditional Random Fields of Ground Motions <i>Robert Jankowski</i>	56
A GIS Based Virtual Urban Simulation Environment <i>Jialiang Yao, Hissam Tawfik, Terrence Fernando</i>	60

Computational Chemistry and Its Applications

Scientific Workflow Infrastructure for Computational Chemistry on the Grid <i>Wibke Sudholt, Ilkay Altintas, Kim Baldrige</i>	69
Application of the Reactivity Index to Propose Intra and Intermolecular Reactivity in Catalytic Materials <i>Abhijit Chatterjee</i>	77
Conformational Processes in L-Alanine Studied Using Dual Space Analysis <i>Chantal T. Falzon, Feng Wang</i>	82
<i>Ab initio</i> Modeling of Optical Properties of Organic Molecules and Molecular Complexes <i>Vladimir I. Gavrilenko</i>	89
A Framework for Execution of Computational Chemistry Codes in Grid Environments <i>André Severo Pereira Gomes, Andre Merzky, Lucas Visscher</i>	97
Thermal Characteristics and Measurement of Nanoscale Materials <i>Taikyeong T. Jeong, Young Seok Song</i>	105
Computational Analysis and Simulation of Vacuum Infusion Molding Process <i>Young Seok Song, Taikyeong T. Jeong</i>	113
Forward, Tangent Linear, and Adjoint Runge-Kutta Methods in KPP-2.2 <i>Philipp Mieke, Adrian Sandu</i>	120
All-Electron DFT Modeling of SWCNT Growth Initiation by Iron Catalyst <i>G.L. Gutsev, M.D. Mochena, C.W. Bauschlicher Jr.</i>	128
<i>Ab initio</i> Study of Chiral Recognition of β -Butyrolactone by Cyclodextrins <i>Waraporn Parasuk, Vudhichai Parasuk</i>	136
C-H Functionalisation Through Singlet Chlorocarbenes Insertions – MP2 and DFT Investigations <i>M. Ramalingam, K. Ramasami, P. Venuvanalingam, V. Sethuraman</i>	143

Theoretical Gas Phase Study of the Gauche and Trans Conformers of 1-Fluoro-2-Haloethanes $\text{CH}_2\text{F}-\text{CH}_2\text{X}$ ($\text{X}=\text{Cl}, \text{Br}, \text{I}$) by Ab Initio and Density Functional Methods: Absence of Gauche Effect <i>Ponnadurai Ramasami</i>	153
Model Dependence of Solvent Separated Sodium Chloride Ion Pairs in Water-DMSO Mixtures <i>A. Asthana, A.K. Chowdhury, A.K. Das, B.L. Tembe</i>	161
Knowledge and Information Management in Computer Communication Systems (KIMCCS 2006)	
Fault Distinguishability of Discrete Event Systems <i>Iwan Tabakow</i>	168
Modelling, Analyzing and Control of Interactions Among Agents in MAS <i>František Čapkovič</i>	176
A Semantic-Driven Cache Management Approach for Mobile Applications <i>Guiyi Wei, Jun Yu, Hanzhao Shi, Yun Ling</i>	184
Fault Tolerance Mechanism of Agent-Based Distributed Event System <i>Ozgur Koray Sahingoz, A. Coskun Sonmez</i>	192
Link Speed Estimation and Incident Detection Using Clustering and Neuro-fuzzy Methods <i>Seung-Heon Lee, M. Viswanathan, Young-Kyu Yang</i>	200
A Consensus-Based Multi-agent Approach for Information Retrieval in Internet <i>Ngoc Thanh Nguyen, Maria Ganzha, Marcin Paprzycki</i>	208
An Adaptive Fuzzy kNN Text Classifier <i>Wenqian Shang, Houkuan Huang, Haibin Zhu, Yongmin Lin, Youli Qu, Hongbin Dong</i>	216
Agent-Based Approach for Distributed Intrusion Detection System Design <i>Krzysztof Juszczyszyn, Ngoc Thanh Nguyen, Grzegorz Kolaczek, Adam Grzech, Agnieszka Pieczynska, Radostaw Katarzyniak</i>	224
A Novel Approach for Similarity Measure Schemes Based on Multiple Moving Objects in Video Databases <i>Choon-Bo Shim, Chang-Sun Shin, DongGook Park, Won-Ho So</i>	232

An Ontology for Network Services <i>Pedro Alípio, José Neves, Paulo Carvalho</i>	240
Contextual Synchronization for Online Co-browsing on Peer-to-Peer Environment <i>Jason J. Jung</i>	244
Modelling of Complex Systems by Cellular Automata (MCSCA 2006)	
Pedestrian Modelling: A Comparative Study Using Agent-Based Cellular Automata <i>Nicole Ronald, Michael Kirley</i>	248
Nagel-Schreckenberg Model of Traffic – Study of Diversity of Car Rules <i>Danuta Makowiec, Wiesław Miklaszewski</i>	256
Path-Planning for Multiple Generic-Shaped Mobile Robots with MCA <i>Fabio M. Marchese, Marco Dal Negro</i>	264
On Modeling and Analyzing Sparsely Networked Large-Scale Multi-agent Systems with Cellular and Graph Automata <i>Predrag T. Tošić</i>	272
Parallel Implementation of a Cellular Automaton Model for the Simulation of Laser Dynamics <i>J.L. Guisado, F. Fernández de Vega, F. Jiménez-Morales, K.A. Iskra</i>	281
Emergent Spatial Patterns in Vegetable Population Dynamics: Towards Pattern Detection and Interpretation <i>Stefania Bandini, Sara Manzoni, Stefano Redaelli, Leonardo Vanneschi</i>	289
Automata Network Simulator Applied to the Epidemiology of Urban Dengue Fever <i>Henrique F. Gagliardi, Fabrício A.B. da Silva, Domingos Alves</i>	297
A Picture for Complex Stochastic Boolean Systems: The Intrinsic Order Graph <i>Luis González</i>	305
Evolutionary Spatial Games Under Stress <i>J. Alonso, A. Fernández, H. Fort</i>	313

Coalescing Cellular Automata <i>Jean-Baptiste Rouquier, Michel Morvan</i>	321
Cellular Automata Architecture for Elliptic Curve Cryptographic Hardware <i>Jun-Cheol Jeon, Kee-Won Kim, Byung-Heon Kang, Kee-Young Yoo</i>	329
Efficient Application of Hybrid 150/90 Cellular Automata to Symmetric Cryptography <i>A. Fúster-Sabater, P. Caballero-Gil, M.E. Pazo-Robles</i>	337
Cellular Automata Preimages: Count and List Algorithm <i>Iztok Jeras, Andrej Dobnikar</i>	345
Self-synchronization of Cellular Automata: An Attempt to Control Patterns <i>J.R. Sánchez, R. López-Ruiz</i>	353
On the Decidability of the Evolution of the Fuzzy Cellular Automaton 184 <i>Angelo B. Mingarelli, Samira El Yacoubi</i>	360
Cell Dormancy in Cellular Automata <i>Mohammad Ali Javaheri Javid, Rene te Boekhorst</i>	367
Dynamic Data Driven Application Systems (DDDAS 2006)	
Introduction to the ICCS2006 Workshop on Dynamic Data Driven Applications Systems <i>Frederica Darema</i>	375
Towards Dynamic Data-Driven Management of the Ruby Gulch Waste Repository <i>Manish Parashar, Vincent Matossian, Hector Klie, Sunil G. Thomas, Mary F. Wheeler, Tahsin Kurc, Joel Saltz, Roelof Versteeg</i>	384
Dynamic Contaminant Identification in Water <i>Craig C. Douglas, J. Clay Harris, Mohamed Iskandarani, Chris R. Johnson, Robert J. Lodder, Steven G. Parker, Martin J. Cole, Richard Ewing, Yalchin Efendiev, Raytcho Lazarov, Guan Qin</i>	393

An Adaptive Cyberinfrastructure for Threat Management in Urban Water Distribution Systems
Kumar Mahinthakumar, Gregor von Laszewski, Ranji Ranjithan, Downey Brill, Jim Uber, Ken Harrison, Sarat Sreepathi, Emily Zechman 401

Model-Driven Dynamic Control of Embedded Wireless Sensor Networks
Paul G. Flikkema, Pankaj K. Agarwal, James S. Clark, Carla Ellis, Alan Gelfand, Kamesh Munagala, Jun Yang 409

WIPER: The Integrated Wireless Phone Based Emergency Response System
Gregory R. Madey, Gabor Szabo, Albert-László Barabási 417

Dynamic Data Driven Application Simulation of Surface Transportation Systems
R. Fujimoto, R. Guensler, M. Hunter, H.-K. Kim, J. Lee, J. Leonard II, M. Palekar, K. Schwan, B. Seshasayee 425

DDAS for Fire and Agent Evacuation Modeling of the Rhode Island Nightclub Fire
Alok Chaturvedi, Angela Mellema, Sergei Filatyev, Jay Gore 433

Auto-steered Information-Decision Processes for Electric System Asset Management
James D. McCalley, Vasant G. Honavar, Sarah M. Ryan, William Q. Meeker, Ronald A. Roberts, Daji Qiao, Yuan Li 440

Data-Driven Power System Operations
E.H. Abed, N.S. Namachchivaya, T.J. Overbye, M.A. Pai, P.W. Sauer, A. Sussman 448

Towards a Dynamic Data Driven System for Structural and Material Health Monitoring
C. Farhat, J.G. Michopoulos, F.K. Chang, L.J. Guibas, A.J. Lew 456

The Omni Macroprogramming Environment for Sensor Networks
Asad Awan, Ahmed Sameh, Ananth Grama 465

Evaluation of Fluid-Thermal Systems by Dynamic Data Driven Application Systems
D. Knight, T. Rossman, Y. Jaluria 473

Inversion of Airborne Contaminants in a Regional Model
Volkan Akcelik, George Biros, Andrei Draganescu, Omar Ghattas, Judith Hill, Bart van Bloemen Waanders 481

Data Assimilation Using the Global Ionosphere-Thermosphere Model <i>I.S. Kim, J. Chandrasekar, A. Ridley, D.S. Bernstein</i>	489
Amplitude-Position Formulation of Data Assimilation <i>Sai Ravela</i>	497
Detection of Tornadoes Using an Incremental Revised Support Vector Machine with Filters <i>Hyung-Jin Son, Theodore B. Trafalis</i>	506
A Generic Multi-scale Modeling Framework for Reactive Observing Systems: An Overview <i>Leana Golubchik, David Caron, Abhimanyu Das, Amit Dhariwal, Ramesh Govindan, David Kempe, Carl Oberg, Abhishek Sharma, Beth Stauffer, Gaurav Sukhatme, Bin Zhang</i>	514
Demonstrating the Validity of a Wildfire DDDAS <i>Craig C. Douglas, Jonathan D. Beezley, Janice Coen, Deng Li, Wei Li, Alan K. Mandel, Jan Mandel, Guan Qin, Anthony Vodacek</i>	522
Development of a Computational Paradigm for Laser Treatment of Cancer <i>J.T. Oden, K.R. Diller, C. Bajaj, J.C. Browne, J. Hazle, I. Babuška, J. Bass, L. Demkowicz, Y. Feng, D. Fuentes, S. Prudhomme, M.N. Rylander, R.J. Stafford, Y. Zhang</i>	530
Blood Flow at Arterial Branches: Complexities to Resolve for the Angioplasty Suite <i>P.D. Richardson, I.V. Pivkin, G.E. Karniadakis, D.H. Laidlaw</i>	538
A New Architecture for Deriving Dynamic Brain-Machine Interfaces <i>José Fortes, Renato Figueiredo, Linda Hermer-Vazquez, José Príncipe, Justin C. Sanchez</i>	546
Dynamically Adaptive Tracking of Gestures and Facial Expressions <i>D. Metaxas, G. Tsechpenakis, Z. Li, Y. Huang, A. Kanaujia</i>	554
Intelligent Management of Data Driven Simulations to Support Model Building in the Social Sciences <i>Catriona Kennedy, Georgios Theodoropoulos</i>	562
Capturing Scientists' Insight for DDDAS <i>Paul Reynolds, David Brogan, Joseph Carnahan, Yannick Loitière, Michael Spiegel</i>	570

An MDA-Based Modeling and Design of Service Oriented Architecture
Adel Torkaman Rahmani, Vahid Rafe, Saeed Sedighian, Amin Abbaspour 578

Advanced Data Driven Visualisation for Geo-spatial Data
Anthony Jones, Dan Cornford 586

Design and Analysis of Test Signals for System Identification
Bo Liu, Jun Zhao, Jixin Qian 593

The Research on the Method of Process-Based Knowledge Catalog and Storage and Its Application in Steel Product R&D
Xiaodong Gao, Zhiping Fan 601

Parallel Monte Carlo Algorithms for Diverse Applications in a Distributed Setting

Small WebComputing Applied to Distributed Monte Carlo Calculations
P.A. Whitlock, Dino Klein, Marvin Bishop 608

Monte Carlo Grid Application for Electron Transport
Emanouil Atanassov, Todor Gurov, Aneta Karaivanova, Mihail Nedjalkov 616

A Monte Carlo Algorithm for State and Parameter Estimation of Extended Targets
Donka Angelova, Lyudmila Mihaylova 624

Error Analysis of a Monte Carlo Algorithm for Computing Bilinear Forms of Matrix Powers
Ivan Dimov, Vassil Alexandrov, Simon Branford, Christian Weihrauch 632

Comparison of the Computational Cost of a Monte Carlo and Deterministic Algorithm for Computing Bilinear Forms of Matrix Powers
Christian Weihrauch, Ivan Dimov, Simon Branford, Vassil Alexandrov 640

International Workshop on Intelligent Storage Technology (IST06)

Performance Analysis of the Cache Conscious-Generalized Search Tree
Won-Sik Kim, Woong-Kee Loh, Wook-Shin Han 648

A Database Redo Log System Based on Virtual Memory Disk <i>Haiping Wu, Hongliang Yu, Bigang Li, Xue Wei, Weimin Zheng</i>	656
Design and Implementation of an Out-of-Band Virtualization System on Solaris 10 <i>Yang Wang, Wei Xue, Ji-Wu Shu, Guang-Yan Zhang</i>	663
High Performance Virtual Backup and Archive System <i>Dan Feng, Lingfang Zeng, Fang Wang, Peng Xia</i>	671
Insurable Storage Services: Creating a Marketplace for Long-Term Document Archival <i>Rahul Simha, K. Gopinath</i>	679
Multi-dimensional Storage QoS Guarantees for an Object-Based Storage System <i>Fei Mu, Jiwu Shu, Bigang Li, Weimin Zheng</i>	687
Design and Implementation of a Random Data-Placement System with High Scalability, Reliability and Performance <i>Kun Liu, Wei Xue, Di Wang, Jiwu Shu</i>	695

Intelligent Agents in Computing Systems

Learning in a Multi-agent System as a Mean for Effective Resource Management <i>Bartłomiej Śnieżyński, Jarosław Kozlak</i>	703
Multicriterial Decision-Making in Multiagent Systems <i>Petr Tučník, Jan Kožaný, Vilém Srovnal</i>	711
JADE-Based A-Team Environment <i>Piotr Jędrzejowicz, Izabela Wierzbowska</i>	719
Agent Factory Micro Edition: A Framework for Ambient Applications <i>C. Muldoon, G.M.P. O'Hare, R. Collier, M.J. O'Grady</i>	727
Crises Management in Multiagent Workflow Systems <i>Małgorzata Żabińska</i>	735
Agent Architecture for Mesh Based Simulation Systems <i>K. Banaś</i>	743

The Application of Agents to Parallel Mesh Refinements in Domain Decomposition Based Parallel Fully Automatic <i>hp</i> Adaptive Finite Element Codes <i>Maciej Paszynski</i>	751
Multiagent Simulation of Physical Phenomena by Means of Aspect Programming <i>Ślawomir Bieniasz, Stanisław Ciszewski, Bartłomiej Śnieżyński</i>	759
Modelling Tactical Driving Manoeuvres with GA-INTACT <i>H. Tawfik, P. Liatsis</i>	767
Agent-Based Mobile Robots Navigation Framework <i>Wojciech Turek, Robert Marcjan, Krzysztof Cetnarowicz</i>	775
The Autonomous Concurrent Strategy for Large Scale CAE Computation <i>P. Uhruski, W. Toporkiewicz, R. Schaefer, M. Grochowski</i>	783
Dynamic Resource Allocation Mechanism for Network Interconnection Management <i>Michał Karpowicz, Krzysztof Malinowski</i>	791
Computing MAS Dynamics Considering the Background Load <i>Maciej Smółka, Robert Schaefer</i>	799
Using Adaptive Agents for the Fault-Tolerant Mobile Computing System <i>Taesoon Park, Jaehwan Youn, Dongryung Kim</i>	807
A Multi-agent Approach to Resource Sharing Optimization in User Networks <i>J.C. Burguillo-Rial, E. Costa-Montenegro, F.J. González-Castaño</i>	815
Heterogeneous Behavior Evaluations in Ethically–Social Approach to Security in Multi-agent System <i>Gabriel Rojek, Renata Cięciwa, Krzysztof Cetnarowicz</i>	823
Semi-elitist Evolutionary Multi-agent System for Multiobjective Optimization <i>Leszek Siwik, Marek Kisiel-Dorohinicki</i>	831
Agent-Based Evolutionary Model for Knowledge Acquisition in Dynamical Environments <i>Wojciech Froelich, Marek Kisiel-Dorohinicki, Edward Nawarecki</i>	839

Quantum-Behaved Particle Swarm Optimization Algorithm with Controlled Diversity <i>Jun Sun, Wenbo Xu, Wei Fang</i>	847
Intelligent Agents as Cells of Immunological Memory <i>Krzysztof Cetnarowicz, Gabriel Rojek, Rafał Pokrywka</i>	855
Negative Selection with Ranking Procedure in Tabu-Based Multi-criterion Evolutionary Algorithm for Task Assignment <i>Jerzy Balicki</i>	863
Multi-objective Optimization Using Co-evolutionary Multi-agent System with Host-Parasite Mechanism <i>Rafał Dreżewski, Leszek Siwik</i>	871
Development of Multi Agent Resource Conversion Processes Model and Simulation System <i>Konstantin A. Aksyonov, Elena F. Smolij, Natalia V. Goncharova, Alexey A. Khrenov, Anastasia A. Baronikhina</i>	879
Designing Floor-Layouts with the Assistance of Curious Agents <i>Ewa Grabska, Katarzyna Grzesiak-Kopeć, Grażyna Ślusarczyk</i>	883
Supporting Software Agents by the Graph Transformation Systems <i>Leszek Kotulski</i>	887
The Outline of the Strategy for Solving Knowledge Inconsistencies in a Process of Agents' Opinions Integration <i>Radosław Katarzyniak, Agnieszka Pieczyńska</i>	891
Agent-Based Service Discovery Middleware in Ubiquitous Environments <i>Hyung-Jun Kim, Kyu Min Lee, Kee-Hyun Choi, Dong Ryeol Shin</i>	895
An Intelligent Middleware Architecture for Context-Aware Service Discovery <i>Kyu Min Lee, Hyung-Jun Kim, Kee-Hyun Choi, Dong-Ryeol Shin</i>	899
Mobile Agent Based Publication Alerting System <i>Ozgur Koray Sahingoz, A. Coskun Sonmez</i>	903
Maintaining Diversity in Agent-Based Evolutionary Computation <i>Rafał Dreżewski, Marek Kisiel-Dorohinicki</i>	908

First International Workshop on Workflow Systems in e-Science (WSES06)

Automatic Transformation from Geospatial Conceptual Workflow to Executable Workflow Using GRASS GIS Command Line Modules in Kepler <i>Jianting Zhang, Deana D. Pennington, William K. Michener</i>	912
A Three Tier Architecture for LiDAR Interpolation and Analysis <i>Efrat Jaeger-Frank, Christopher J. Crosby, Ashraf Memon, Viswanath Nandigam, J. Ramon Arrowsmith, Jeffery Conner, Ilkay Altintas, Chaitan Baru</i>	920
Workflows for Wind Tunnel Grid Applications <i>A. Paventhan, Kenji Takeda, Simon J. Cox, Denis A. Nicole</i>	928
Distributed Execution of Workflows <i>Ismael Navas-Delgado, Jose F. Aldana-Montes, Oswaldo Trelles</i>	936
Applying Workflow to Experiment Control in Virtual Laboratory <i>Lukasz Czekierda, Krzysztof Zieliński</i>	940
Integration of Compute-Intensive Tasks into Scientific Workflows in BeesyCluster <i>Pawel Czarnul</i>	944
A Distributed Re-configurable Grid Workflow Engine <i>Jian Cao, Minglu Li, Wei Wei, Shensheng Zhang</i>	948
Adding Instruments and Workflow Support to Existing Grid Architectures <i>D.J. Colling, L.W. Dickens, T. Ferrari, Y. Hassoun, C.A. Kotsokalis, M. Krznaric, J. Martyniak, A.S. McGough, E. Ronchieri</i>	956
Workflow Deployment in ICENI II <i>A. Stephen McGough, William Lee, John Darlington</i>	964
Agent-Based Middleware Architecture for Workflow in Grid Portals <i>Sangkeon Lee, Jaeyoung Choi, Keumwon Cho</i>	972
Cooperative Processes for Scientific Workflows <i>Khaled Gaaloul, François Charoy, Claude Godart</i>	976

Semantic Tools for Workflow Construction <i>Ondrej Habala, Marian Babik, Ladislav Hluchy, Michal Laclavik, Zoltan Balogh</i>	980
Stochastic Modeling and Quality Evaluation of Workflow Systems Based on QWF-Nets <i>Yunni Xia, Hanpin Wang, Chunxiang Xu, Liang Li</i>	988
Styx Grid Services: Lightweight, Easy-to-Use Middleware for Scientific Workflows <i>J.D. Blower, A.B. Harrison, K. Haines</i>	996
Automatic Services Composition in the Grid Environments <i>Wenju Zhang, Fei Liu, Shudong Chen, Fanyuan Ma</i>	1004
A Non-intrusive and Incremental Approach to Enabling Direct Communications in RPC-Based Grid Programming Systems <i>Alexey Lastovetsky, Xin Zuo, Peng Zhao</i>	1008
Enacting Proactive Workflows Engine in e-Science <i>Ezio Bartocci, Flavio Corradini, Emanuela Merelli</i>	1012
Networks: Structure and Dynamics	
Traffic Noise and Maximum-Flow Spanning Trees on Growing and Static Networks <i>Bosiljka Tadić, Stefan Thurner</i>	1016
Local Information Based Algorithms for Packet Transport in Complex Networks <i>Bernard Kujawski, G.J. Rodgers, Bosiljka Tadić</i>	1024
Empirical Analysis of the Spatial Genetic Algorithm on Small-World Networks <i>Yong Min, Xiaogang Jin, Xianchuang Su, Bo Peng</i>	1032
An Evolution Process Model for the Internet Topology <i>Sangjoon Park, Insook Cho, Byunggi Kim</i>	1040
Attack Strategies on Complex Networks <i>Lazaros K. Gallos, Reween Cohen, Fredrik Liljeros, Panos Argyrakis, Armin Bunde, Shlomo Havlin</i>	1048
Elementary Modules in Games Networks <i>Matthieu Manceny, Franck Delaplace</i>	1056

A New Analysis Method for Complex Network Based on Dynamics of Spin Diffusion <i>Makoto Uchida, Susumu Shirayama</i>	1063
Simulation of Micro-, Grand-, and Canonical Ensembles of Complex Networks <i>Christoly Biely, Stefan Thurner</i>	1067
Synchronization in Network Structures: Entangled Topology as Optimal Architecture for Network Design <i>Luca Donetti, Pablo I. Hurtado, Miguel A. Muñoz</i>	1075
Dynamics of Content-Based Networks <i>Duygu Balcan, Ayşe Erzan</i>	1083
Social Connections and Access Charges in Networks <i>Rodrigo Harrison, Gonzalo Hernandez, Roberto Munoz</i>	1091
Topology of Cell-Aggregated Planar Graphs <i>Milovan Šuvakov, Bosiljka Tadić</i>	1098
Geographical Construction of Scale-Free Networks with Both Short Path Lengths and Hops <i>Yukio Hayashi, Jun Matsukubo</i>	1106
Collaborative Tagging as a Tripartite Network <i>Renaud Lambiotte, Marcel Ausloos</i>	1114
Author Index	1119

Information Registry of Remotely Sensed Meta-module in Grid Environment

Yong Xue^{1,3}, Jianqin Wang^{2,*}, Chaolin Wu¹, Yincui Hu¹, Jianping Guo¹, Lei Zheng¹,
Wei Wan¹, Guoyin Cai¹, Ying Luo¹, and Shaobo Zhong¹

¹ State Key Laboratory of Remote Sensing Science, Jointly Sponsored by the Institute of Remote Sensing Applications of Chinese Academy of Sciences and Beijing Normal University, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P.O. Box 9718, Beijing 100101, China

² College of Information and Electrical Engineering, China Agricultural University, P.O.Box 142, Beijing, 10083, China

³ Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB, UK
Tian1.wang@163.com, y.xue@londonmet.ac.uk

Abstract. The term "the Grid" was coined in the mid 1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. Grid computing technology is a new way for remotely sensed data processing. Special remotely sensed data processing Grid node provides convenient, consistent spatial information processing service for user. In this paper we mainly discusses information registry mechanism and method for remotely sensed data processing module and algorithm in Grid environment and gives the remotely sensed meta-module conception. Then based on the characteristic of remotely sensed data processing module and Grid technology, we describe remotely sensed data processing meta-module information registry method and give a concrete example.

1 Introduction

Grid is an integrated computing and resources environment, or a location for computing resources. Grid can absorb all kinds of resources for computing, and transfer them into available, reliable, standard and economic computing ability. Besides computers, computing resources also include web communication abilities, data, instruments and equipments, and even people. Foster *et al.* [4] defined "Grid problem", which is defined as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources –what they referred to as virtual organizations (VO) [3].

Dozens of satellites constantly collecting data about our planetary system 24 hours a day and 365 days a year [7]. Remote Sensing data is characterized by largeness and instantaneousness. The analysis and sharing of these huge amounts of data is a big

* Corresponding author.

challenge for the remote sensing community [6]. Grid computing technology provides a good way to process remotely sensed data and sharing the processed spatial information through combining virtual super computing power storage resource on the Internet.

Today there have several famous Grid projects in spatial information field. Within the DataGrid project funded by the European Union, Work Package (WP) 9 of the Data aims to demonstrate the use of Grid technology for remote sensing applications and earth observation [5]. Earth Observation Grid (ESG) is a research project sponsored by the U.S. DOE Office of Science under the Scientific Discovery through Advanced Computing program (SciDAC). The aim of this project is to analyze climate variety all over the earth surface [1]. Information Power Grid (<http://www.ipg.nasa.gov/>) is the NASA's computational support for aerospace development, planetary science and other NASA research.

The spatial information can be collected, shared, organized, and processed with Spatial Information Grid (SIG), which has been sponsored by Ministry of Science and Technology of the People's Republic of China. It is the infrastructure that manages and processes spatial data according to users' demand (<http://www.spatial.org>). Through demonstration applications in some key fields, develop Spatial Information Grid to promote the industrialization of spatial information resources. ([http:// www .863.org.cn/english/annual_report/annual_repor_2001/200210090014.html](http://www.863.org.cn/english/annual_report/annual_repor_2001/200210090014.html))

Remotely sensed data is one of the most important spatial information sources, so the research on architectures and technical supports of Grid-based remotely sensed data processing node is the significant part of the research on SIG. Today, remotely sensed data processing modules and algorithms are the valuable resources, which can be used as a service for users who are not familiar with remotely sensed data processing methods. It can also avoid rewriting the same algorithms over and over again for different people through sharing the remote sense algorithms and module as a resource service. The key of using this valuable resource is how to let users conveniently find them and how to get use of them in Grid environment in order to share resources. To solve the above problem, we must let the remotely sensed data processing module be indexed in the Grid environment. The service information of the remotely sensed data processing module should clearly describe category of the algorithms and how to operate in Grid environment with these resources. Grid service request will be submitted to special remotely sensed data processing node with virtual super computing power after user searching the service information. Service Data technology of OGSA in Globus provides a effective method for information registry. We will discuss the metadata and method and mechanism of remotely sensed data processing module service registry in Grid environment based on Globus toolkit3.0.2 as follows.

2 Grid-Based Remotely Sensed Meta-data Service Data

Service Data technology of Globus (GT3.x) provides the fundamental method to describe service content in Grid environment.

2.1 Service Data of OGSA (Open Grid Service Architecture)

Service Data is the aggregation structure that is related to Grid service [2]. It must be conveniently indexed and classified through Service Data Element (SDE). A service data declaration is a mechanism of publicly expressing the available state information of a service through a known schema. There are two types of service data elements (<http://www-128.ibm.com/developerworks/grid/library>):

- **Static:**
Declared as part of the service's interface definition (GWSDL portType definition)
- **Dynamic:**
Added to a service instance dynamically. This behaviour is implementation-specific. The client may know the semantics (type and meaning) of the service data, or you can acquire that information from somewhere. For example, in order to process the dynamic SDE values you may need to get the schema for the SDE from a remote location.

The Service Data Element (SDE) model provides standard mechanisms for querying, updating, adding and removing data associated with each Grid service instance

2.2 Remote Sensing Meta-module Service Data in Grid Environment

2.2.1 Conception of Grid-Based Remote Sensing Meta-module

We give the conception of Grid-based remote sensing meta-module as following: The metadata that describe serials of remotely sensed data processing modules and algorithms in order to publish and manage these resources in Grid environment. The description of Grid-based remote sensing meta-module must include function express of the algorithm, input and output description, control parameter description etc. As Grid-based remote sensing meta-module, they must match these conditions:

- They must be described as the form of the Service Data Element (SDE) of Grid Service.
- Expression schema must match with the rule of the Grid Web Service Description Language (GWSDL)
- They can clearly describe the name, function, input and output parameter of the remotely sensed data processing module.
- Service Data used to describe remote sensing module must be static, which will not change until the module and algorithms become update.

2.2.2 Meta-module Service Data Structure in OGSA

The structure of the remote sensing meta-module seems as dendriform organization (Figure 1). RSService Instance will be produced by remotely sensed data processing service factory. We can give above figure embodiment: RSService Factory can be supposed as an service factory to produce remotely sensed image classification service instance. The same classification algorithm has different Service Data Set, which is made up of several Service Data Elements. Also we can suppose that Service Data Element A describes that classification algorithm service can provide classified

image after the task, and Data Element B describes that classification service can provide description document or log file about this task.

If users not only want to get the classified image but log files about this task, they can choice RSService Instance A that is described by Service Data Set A. If users only want to get the classified image, they can just choice RSService Instance B that is described by Service Data Set B.

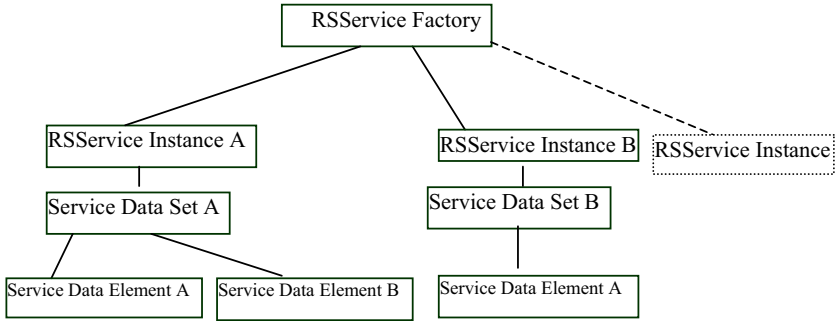


Fig. 1. Remote sensing Service Data structure

2.2.3 Grid-Based Remotely Sensed Meta-module Service Data Item

In order to clearly describe remotely sensed data processing module service information, Grid-based remote sensing meta-module Service Data must include description of the module and algorithm, input data, threshold value, output information, running time. Service Data items are shown in the following table (Table1). Keyword item mainly contains description content of Grid-based meta-module. Model_Description item mainly express the name of the algorithm, function, developed date of the module. Input_Data item seems complex because of diversity of the remotely sensed data format. Output_Data item has the flexibility according to user’s demand.Remote sensing information service can provide image or document result. Data Type item is the XML element type description in Grid Web Services Description Language (GWSDL).

Table 1. Item of Grid-based remote sensing meta-module Service Data

	Keyword		Data Type	Note
Description	Model_Description		Description	Whole description
Input	Input_Data		GridRS_Meta	Input data format
Threshold	Threshold_Con		xs:float	Threshold and control parameter
Output	Output_Data	Image, Document	xs:string	Output format
Running time	RunTime_MByte		xs:duration	Seconds for per Mbyte

RunTime_Mbyte means how many seconds this module process per mega-byte remotely sensed data using a certain algorithm. Though this service data item ,user can easily estimate the time by simple multiplication because most of the algorithm analyze pixel one by one. But the result of RunTime_Mbyte must be tested before a meta-module service data being published. There are many remote sensing data types which is described as complex type in Grid Web Service Description Language(GWSDL). Input_Data Service Data is the description of remote sensing data needed by the algorithm and module in the following table (Table 2).

Table 2. Item of Input_Data Service Data

Data Format	XML ComplexType	Data Type	Note
GridRS_Meta	Data_Quality	xs:int	Full cloud covered is the max value(100) 0-100 scale
	Data_Format	xs:string	Data format name
	Imaging_Time	xs:dateTime	Collect data time

2.2.4 Service Data Description of Grid-Based Remotely Sensed Meta-module

Service Data Description (SDD) can be used to express remote sensed meta-module Service Data. Remotely Sensed meta-module SDD XML Schema is made up of six parts: domain name, version copyright, module information, module algorithm and running time, input information, output information.

<?The first part: domain definition>

```
...
<wsdl:definitions name=" ModelData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
...
<wsdl:types>
```

<?The Second part: copyright statement>

```
...
```

<?The third part: Information description of module>

```
<complexType name=" ModelDataType ">
<sequence>
<element name="Model_Description" type="ModelDescription"/>
<element name="Input" type="InputInfo"/>
<element name="Output" type="OutputInfo"/>
```

```
...
```

<?The fourth part: algorithm and time>

```
<complexType name=" ModelDescription ">
<sequence>
<element name="ArithmeticName" type="string"/>
<element name="Function " type="string"/>
<element name="ModelDate " type="date"/>
</sequence>
</complexType>
```

```

<?The fifth part: input information>
  <complexType name=" InputInfo ">
    ...
  <complexType>
    <element name=" format " type=" string "/>
    <element name=" imagingDate " type=" dateTime "/>
    <element name=" imageQuality " type="ImageQuality "/>
  <element name="ImageQuality" >
  <sequence>
  <element name="cloud" type="int" minOccurs="0" maxOccurs="1"/>
    ...
<?The sixth part: output information>
  <complexType name=" OutputInfo" type="OutputData">
    ...
  <element name="ImageFormat" type="string"/>
  ...
</wsdl: definitions>

```

3 Grid-Based Remotely Sensed Meta-module Service Registry

Combing Globus Open Grid Service Infrastructure (OGSI) rules and remotely sensed meta-module above, we provide the remotely sensed Grid service registry method (Figure 2). Serials of remotely sensed data processing module service data is registered in the registry container of the special remotely sensed data processing Grid node through Service Data Provider, such as NDVI Service Data Provider, aerosol computing Service Data Provider, image classification Service Data Provider, etc. For user, they can search the proper service from the service data through special portal made up of remote sensing service set. There are two ways to find service through Grid service, one is based on demand line, and the other is Grid browser (Figure 3). User can use the browser to find the proper service description while inputting the service Uniform Resource Locators (URL) of remotely sensed data processing service.

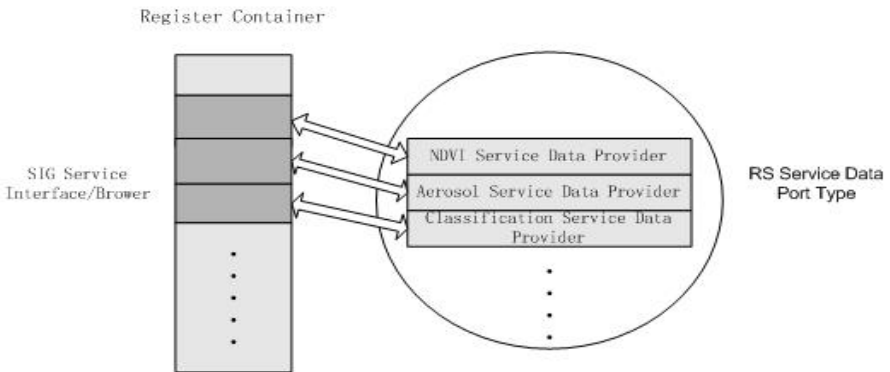


Fig. 2. Grid-based remotely sensed meta-module registry process

Name	PortType	Type	Min	Max	Mutability	Modifiable	Nillable
serviceData...	{http://www.grid...	{http://www.w3...	0	unbounded	mutable	false	false
factory_ocstor	{http://www.grid...	{http://www.gr d...	1	-	mutable	false	true
gridServiceHan...	{http://www.grid...	{http://www.gr d...	0	unbounded	extendable	false	false
gridServiceRef...	{http://www.grid...	{http://www.gr d...	1	unbounded	mutable	false	false
findServiceDat...	{http://www.grid...	{http://www.gr d...	1	unbounded	static	false	false
setServiceData...	{http://www.grid...	{http://www.gr d...	1	unbounded	static	false	false
terminatorTime	{http://www.grid...	{http://www.gr d...	1	-	mutable	false	false
membershipC...	{http://www.grid...	{http://www.gr d...	1	unbounded	constant	false	false
entry	{http://www.grid...	{http://www.gr d...	0	unbounded	mutable	false	false
notifiableSeric...	{http://www.grid...	{http://www.w3...	0	unbounded	mutable	false	false
subscribeExten...	{http://www.grid...	{http://www.gr d...	1	unbounded	static	false	false

```

ns2:serviceData/values xmlns:ns2="http://www.gridforum.org/namespaces/2003/03/serviceData"
ogsi:subscribeExtensibility inputElement="ogsi:subscribeByServiceDataName" xmlns:ogsi="http://www.gridforum.org/names

```

Fig. 3. Grid-based remotely sensed processing service searching through browser

4 Conclusion

Grid technology is a very effective method for remotely sensed data processing service. Through it the existing remotely sensed data processing modules and algorithms resource can be shared seems like the common service in Grid environment. It much enhances the resource utility rate. In order to publish, describe, and manage Grid service data, the paper puts forward a Grid-based remotely sensed meta-module conception and corresponding structure. Combing the rule of this conception and OGSi technology, we define the Grid-based remotely sensed meta-module service data and Service Data Description language schema. Through remotely sensed Service Data Provider, the remote sensing service can be register in the Registry Container of the special remotely sensed data processing Grid node. Users can also search and choice the proper service by demand line or Grid browser.

After that, he can send a service request to Grid computing node according to the service data description.

Acknowledgement

This publication is an output from the research projects "Grid platform based aerosol fast monitoring modeling using MODIS data and middlewares development" (40471091) funded by National Natural Science Foundation of China (NSFC), China, "Dynamic Monitoring of Beijing Olympic Environment Using Remote Sensing" (2002BA904B07-2) and "863 Program - Remote Sensing Information Processing and Service Node" (2003AA11135110) funded by the MOST, China, "Digital Earth" (KZCX2-312) funded by Chinese Academy of Sciences, China, and "Research Fund for Talent Program" funded by China Agricultural University.

References

- [1] Allcock, B. I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Leigh, A. Sim, A. Shoshani, B. Drach, D. Williams. SC 2001. High-Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies., November 2001.
- [2] Chen, D., Demichev, A., Foster, D., Kalyaev, V., Kryukov, A. et al. OGSA Globus Toolkit3 evaluation activity at CERN Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 534, Issues 1-2, 21 November 2004, Pages 80-84.
- [3] Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
- [4] Foster, I. C. Kesselman, S. Tuecke *The Anatomy of the Grid, Intl J. Supercomputer Applications*, 2001.
- [5] Giovanni, N. A., Luigi, F. B., and Linford, J., 2003, Grid technology for the storage and processing of remote sensing data: description of an application. In *Proceedings of the society of photo-optical instrumentation engineers (SPIE)*, Vol. 4881, 677-685, 2003.
- [6] Hu Yincui, Xue Yong, Tang Jiakui, Zhong Shaobo, Cai Guoyin, 2005, Data-parallel Georeference of MODIS Level 1B Data Using Grid Computing. *Lecture Notes in Computer Science*, Vol. 3516, pp883-886.
- [7] Wang, J. Q., Xue, Y., and Guo, H. D., 2003, A Spatial Information Grid Supported Prototype Telegeoprocessing System. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS'2003) held in Toulouse, France on 21-25 July 2003*.
- [8] Xue Yong and Wang Jianqin, Wang, Y. G., Wu, C. L. and Hu, Y. C., 2005, Preliminary study of Grid computing for remotely sense information. *International Journal of Remote Sensing*, Vol.26, No.16, pp3613-3630.

Preliminary Study of Avian Influenza A Infection Using Remote Sensing and GIS Techniques

Jianping Guo^{1,4}, Yong Xue^{1,2,*}, Shaobo Zhong^{1,4}, Chunxiang Cao¹, Wuchun Cao^{3,*}, Xiaowen Li¹, and Liqun Fang³

¹ State Key Laboratory of Remote Sensing Science, Jointly Sponsored by the Institute of Remote Sensing Applications of Chinese Academy of Sciences and Beijing Normal University, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P.O. Box 9718,

Beijing 100101, China

² Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB, UK

³ Institute of Microbiology and Epidemiology, Academy of Military Medical Sciences, Chinese PLA, Beijing 100071, PR China

⁴ Graduate School of the Chinese Academy of Sciences, Beijing, China
gjppgis@163.com, y.xue@londonmet.ac.uk

Abstract. The outbreak of Avian Influenza A (H5N1) infection has spread across all over the world from East-South Asia to Russia, Greece, Romania and Turkey. It will be important to find the transmission route and determine the environmental factor that affect the prevalence of avian influenza A virus. Based on the environmental parameters derived from remote sensing (RS) measurements and the avian influenza A (H5N1) infection case data in China during January 23, 2004 to February 24, 2004, the correlations between the outbreak of H5N1 avian influenza and the environmental parameters of the infected area, such as land surface temperature, was conducted using the spatial analysis abilities of GIS. The statistically significant association between the land use or land cover and outbreak of avian influenza A infection was found, i.e. about 86.4% of the 44 cases are in the cropland. Besides, by the buffering analysis, it is estimated that the vicinity at 50 km or so to main railways plays a key role in the spatial distribution of avian influenza A infection. Finally, we draw preliminary conclusion that the infection often outbreak in a certain range of land surface temperature etc probably due to in part the H5N1 virus implications.

1 Introduction

The avian influenza A infection has outbroken in poultry and wild birds around the world, killing thousands of poultry, which has already aroused much attention from nearly all the people. According to WHO latest information, the human cases have amounted to 118 infected, included 61 dead since December 1997 [2]. As for as avian influenza A is concerned, we only know that the virus, particularly the high pathogenic

* Corresponding authors.

H5N1 strain, can easily mutate to transmit to human directly, if appropriate condition is satisfied. However, from the perspective of epidemiology and effectively control, the transmission, the origin, the ecosystem of the outbreak and spatial pattern of avian influenza A infection outperform above-mentioned factors, which is particularly in the interests of many environmental epidemiologist.

Geographic Information Systems (GIS) and remote sensing (RS) technologies are being used increasingly to study the spatial and temporal patterns of infectious diseases (Brooker and Michael, 2000), which show great potential to serve as: (1) an effective data capture, mapping and analysis tool for the development of spatial epidemiological diseases; (2) an environment for modeling the spatial distribution of infection accounting for the RS derived parameters and climate measures; and (3) a focal tool in infection control given their abilities to better define the endemic area and predict precisely the risk of the population exposed to some infections.

In this paper, the data and methods used in the study were introduced in Section 2. The analysis of data was conducted in Section 3. Finally, we gave the preliminary conclusions.

2 Data and Methods

2.1 Pre-processing of Infection Case Data

The first avian influenza outbreak in the Guangxi Zhuang autonomous region in Southwestern China on January 23, 2004. From then on till February 24, 2004, there were other 43 cases in total of avian influenza A infection across China, which were all high pathogenic i.e. H5N1 strain virus. It is very uncommon in China for so many cases in such short time. We chose the case data during the period as the objective of our study concerning the statistical implications. Accounting for the time lags between report of suspected case and determining of case, the time of case outbreak was determined on the day of report of suspected case, which is better in line with the true situation of avian influenza A infection. The spatial coordinate of avian influenza A infection was centred at the county level. The spatial resolution of the analysis was based on the 1km x 1km pixel.

2.2 Preparation of Land Use/ Land Cover Data

Due to the little change of land use/land cover, we chose the MODIS land cover product (MOD12Q1) of 2001 in China, which identifies 17 classes of land cover including 11 natural vegetation classes, 3 developed land classes (1 class of which is a mosaic with natural vegetation), 1 permanent snow or ice class, 1 barren or sparsely vegetated class, and 1 water class.

2.3 Land Surface Temperature Measurement

We chose MODIS thermal infrared bands 1B data to derive land surface temperature using the split-window technique [3]. In order to reduce the random error, we derived the mean temperature of the infected area by averaging the measures of neighbouring 8 cells (1km x 1km).

3 Analysis

Buffering is a process of identifying objects within a specified distance of a reference object. A simple environmental example would be to create a buffer around the strip. In regarding to the avian influenza, Buffer could be used to assess potential risks to the affected avian population. In this case, we built a buffer around the main railway of China (Figure 1) on both sides by 50 kilometres away the main railway of China. Based on the buffer analysis, there are 43 cases (97.8%) in the buffer area, which suggests that some strong association between the railway and the outbreak of avian influenza A infection exists.

From the mean land surface temperature derived from MODIS data, we got a plot of temperature

range in 44 cases of avian influenza A infection in China, from which we found that about 66% (29/44) of avian influenza cases outbreak in the range of (7°C , 13°C), other 34% in the range of ($-13^{\circ}\text{C} - 7^{\circ}\text{C}$) and ($17^{\circ}\text{C} - 27^{\circ}\text{C}$). It is very important for us to effectively prevent and control the avian

influenza for temperature varies in the ($7^{\circ}\text{C} - 13^{\circ}\text{C}$) range in the autumn and spring in most part of China. So we should take any measures to strengthened monitoring of the avian influenza prevalence in the season when temperature varies in that range.

Land cover type is very popular in the environmental epidemiology, especially for vector-borne infection disease, because the vector habitat usually prefers some specific land cover type to other types. Meanwhile, there are great evidences of avian influenza transmitted from migratory birds, which often fly through some specific routes. Therefore, when we tried to find the environmental factor influencing the outbreak of avian influenza A infection, we incorporated the land cover type into our research. Through overlay analysis of case data and land cover data, we find that there are 28 cases (64%) outbreak in cropland.

4 Conclusion

Besides the land cover type, we also got some association between mean land surface temperature and avian influenza, from the perspective of cluster in the specific range of mean land surface temperature in the area affected by avian influenza A infection.

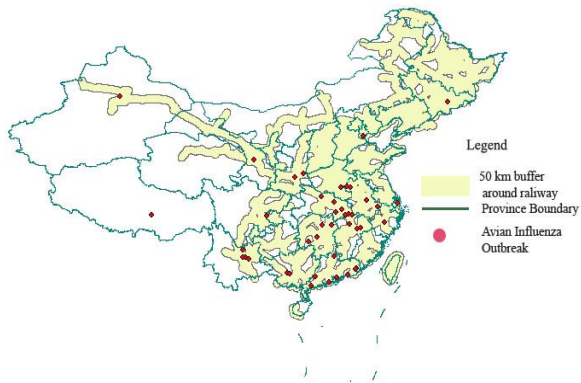


Fig. 1. The 50 km buffer map around main railway network of China, which is overlaid with avian influenza A infections cases of the period between January 23- February 24, 2004

This can be implicated that avian influenza A (i.e. H5N1) virus often adapts to some range of temperature, which is in line with the high frequency of avian influenza in winter and spring when the land surface temperature in most of China accords to the range calculated in our study. However, accounting for the avian influenza prevalence in May 2005, a plenty of migratory birds died of H5N1 strain avian influenza in Qinghai Lake, and the prevalence in the southeast Asia, where the land surface temperature often lies out of the range we got in the study. In the study, strong association as well between railway and avian influenza A infection holds true. This can be explained by the fact that in China, the transportation by railway of poultry products is the key way besides the highway for most poultry managers. In this study, we only investigated the correlation between railway and avian influenza A infection.

Acknowledgement

This publication is an output from the research projects “Monitoring of Beijing Olympic Environment Using Remote Sensing” (2002BA904B07-2) funded by the MOST, China, "Digital Earth" (KZCX2-312) funded by CAS, China. This study was also funded by NSFC, China (grant number: 30590370, 30590374) and national military of sciences and technologies of China (grant number: 2004BA519A32).

References

- [1] Brooker S and Michael E.: the potential of geographical information systems and remote sensing in the epidemiology and control of human helminth infections. *Advances in Parasitology*, edited by Hay S. I., Randolph S. E., and Rogers D. J. (Academic Press) Vol.47 (2000).
- [2] Tran Tinh Hien, M.D., Nguyen Thanh Liem, M.D., Nguyen Thi Dung, M.D., Luong Thi San, M.D., Pham Phuong Mai, M.D., *et al.*: Avian influenza A (H5N1) in 10 patients in Vietnam. *England Journal of Medicine*, 350 (2004) 1179-88.
- [3] Xue Y., Cai G.Y., Guan Y. N., and Tang J. K.: Iterative self-consistent approach for earth surface temperature determination. *International Journal of Remote Sensing*, 26(1), (2005) 185-192.

Efficient Coding of Quadtree Nodes

Mariano Pérez, Xaro Benavent, and R. Olanda

Instituto de Robótica. University of Valencia
Polígono de la Coma, s/n.
Aptdo. 22085, 46071-Paterna, Spain
`Mariano.Perez@uv.es`

Abstract. In this paper an alternative non-pointer quadtree node codification to manage geographical spatial data is presented. New codification is based on a variable sequence of z-ordered base four digits. Memory requirements of the new codification are lower than previous codifications, and in particular lower than FD codification, the most commonly used in linear quadtrees. Furthermore, z-ordering makes compatible new codification with most of the algorithms developed for FD.

1 Introduction

In recent years, a wide set of data structures has been developed to manage spatial data in Geographic Information Systems (GIS) field. The most popular approaches are based on quadtrees, bintrees, R-trees, the cell tree and the grid file [3]. The quadtree is probably one of the most extended for spatial information representation, and amongst the different kinds of quadtrees, the region quadtree is the most used [1].

Region quadtrees can be represented in memory as a dynamic tree (using pointers) or they can be stored using a set of linear codes (without using pointers). Pointer-based approach is ill-suited for implementing disk-based structures whereas linear codes approach is especially interesting when the quadtree size is quite large or when it is stored on secondary storage devices. Memory requirements in the case of linear codes is lower than when using pointers, but it is possible to regenerate the whole structure from only a few leaf nodes, as it happens for linear quadtrees [1].

2 Previous Quadtree Codifications

There are several ways of implementing linear codes that are able to identify the quadtree nodes. Usually the codification employs numbers in base four or five, called locational codes. The digits of these numbers are directional codes which locates the node within the quadtree from the root node. Depending on the implementation, apart from the location code, the node level (depth of the leaf node within the quadtree) is also stored. The most popular linear implementations are the FL (Fixed Length), the FD (Fixed Depth) and the VL (Variable Length) [3].

The FL and VL locational codes are based on a base five sequence of digits. Each digit indicates NW, NE, SW, SE and "don't care" directions. These codifications do not need an additional field to codify the node level. This information is implicit in the directional code; the "don't care" value indicates the node is a leaf. The main difference between VL and FL is that the VL locational codes have a variable length and it is usually shorter than the FL locational one.

The FD codification is usually implemented as a structure with two fixed size fields. The first field, called quadcode [2], stores the directional code and indicates the position and orientation of the node within the quadtree. The second field indicates the node level.

The quadcode is a z-ordered base four code and consists of a sequence of values:

$$q = q_1 q_2 \dots q_N \quad (1)$$

where $q_i = 0, 1, 2, 3$, with $i = 1, 2, \dots, N$, being N the code length (which is the maximum number of tree levels). For each square block in which a quadrant is divided, a new value q_i is added to have a unique characterization of each node.

Among the three indicated codifications, the FD is the most referenced in scientific literature because the FD locational code (quadcode) is a base four code, and it is easily handled using two bits per digit, which allows the use of binary and logical operations to carry out the codification and de-codification processes. Furthermore, FD codification is the most used in linear region quadtrees [4].

3 Base 4 Variable Length Code (B4VL)

B4VL is an improved alternative version of the FD codification, where the locational code and the node level are integrated in a unique code, and where locational code has variable size. These properties are not present in the FD codification.

To carry out the packing process in B4VL, the bits that define the code are divided into two parts. The first part, corresponding to low bits, has a fixed size and is used to store the node level. The other part has a variable length, and stores the locational code (Figure 1).

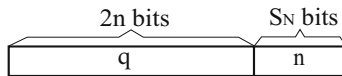


Fig. 1. Bits distribution in B4VL codification. First part correspond to node level (n). Second part correspond to locational code (q).

The B4VL locational code is similar to quadcode (equation 1), but its size is optimized. If n is the node level, the locational code is a length n sequence of values:

$$q = q_1 q_2 \dots q_n \quad (2)$$

where $q_i = 0, 1, 2, 3$, being $i = 1, 2, \dots, n$.

If the maximum node level in the quadtree is N , the number of bits used to codify the first part of the node code is $\lceil \log_2(N + 1) \rceil$, while the second part needs $2n$ bits, being n the node level. The length of the first part is independent of the node level, so it is only computed once and this value is stored in memory. The number of bits of the second part is variable and equal to twice the node level (see Figure 1).

The algorithm that generates the code from the node level (n) and the locational code (q) is based on the following equation:

$$code \leftarrow ((q \ll S_N) + n) \quad (3)$$

where \ll is the left bitwise binary operator, and S_N is the number of bits in the first part of the code. This number is the same for every node in the quadtree and its value is $\lceil \log_2(N + 1) \rceil$, where N is the maximum node level in the quadtree.

The inverse operation is based on the following equations:

$$\begin{cases} n \leftarrow (code \& M_S) \\ q \leftarrow (code \gg S_N) \end{cases} \quad (4)$$

where $\&$ is the AND binary operator, and \gg is the right bitwise operator; M_S represents a binary mask computed once using the equation: $M_S = 2^{S_N} - 1$.

Both operations (equations 3 and 4) have an optimal computational cost, which is constant ($O(1)$) with node level.

4 Results

As we have indicated above, FD and FL codes have a fixed size, while VL and the proposed B4VL codification have a variable size depending on the depth of node. Table 1 shows the storage cost for the four codifications. The first part of the summation for the B4VL and the FD codifications is related to the number of bits used to store the node level and the second part to the quadcode.

Table 1. The number of bits used in B4VL, FD, FL and VL codifications based on node level n (depth) and based on the maximum achieved level of the quadtree N

B4VL	$E(n) = \lceil \log_2(N + 1) \rceil + 2n$
FD	$E(n) = \lceil \log_2(N + 1) \rceil + 2N$
FL	$E(n) = \lceil N \cdot \log_2 5 \rceil$
VL	$E(n) = \lceil (n + 1) \cdot \log_2 5 \rceil$ (if $n < N$) OR $E(n) = \lceil (N) \cdot \log_2 5 \rceil$ (if $n = N$)

In order to compare the memory space required for each kind of codification, the average number of bits needed to codify the nodes included in fully occupied quadtrees will be considered (quadtrees are selected based on its maximum level N). Results are shown in Table 2.

Table 2 highlights that B4VL has variable memory size, but it is always lower than the space required by FD. It also shows that B4VL size codes are higher than FL and VL for trees of small height, but they are much lower for deeper trees ($N \geq 7$).

Table 2. Average node bits for each codification based on the maximum quadtree node level N

N	1	2	3	4	5	7	10	13	17	20	25	30	40
B4VL	3.00	5.60	7.43	10.36	12.34	16.33	23.33	29.33	38.33	44.33	54.33	64.33	85.33
FD	3.00	6.00	8.00	11.00	13.00	17.00	24.00	30.00	39.00	45.00	55.00	65.00	86.00
FL	3.00	5.00	7.00	10.00	12.00	17.00	24.00	31.00	40.00	47.00	59.00	70.00	93.00
VL	3.00	5.00	6.91	9.80	11.83	16.77	23.77	30.77	39.82	46.82	58.77	69.83	92.83

5 Conclusions

In this paper a new approach to codify nodes in quadtree representation has been presented. The approach is based on a unique compact base four representation of the locational code and the level of the node.

It has been demonstrated that the storage cost of the proposed codification is better than previous codifications (table 2). Space memory requirements for B4VL codification are always lower than the memory needed for FD codification. Compared to FL and VL codifications, B4VL is similar for small height quadtrees and lower for deeper quadtrees, but B4VL uses a base four digits code, so its coding and decoding process is much more simple and efficient than FL and VL.

An important property is that the sequence of relevant bits in B4VL locational code (equation 2) and in FD locational code (equation 1) are equivalent in practice. So, it is possible that algorithms developed for FD works with B4VL making minor modifications.

Acknowledgements. This work has been supported by the project UV-AE-20050985 of the University of Valencia and the project GV2005-184 of the Generalitat Valenciana.

References

1. Gargantini, I.: An Effective Way to Represent Quadtrees. Communications of the ACM. 25(12), 1982, pp. 905–910.
2. Li, S., Loew, M. H.: The Quadcode and its Arithmetic. Communications of the ACM. 30(7), 1987, pp. 621–626.
3. Samet, H.: Applications of spatial data structures: Computer graphics, image processing, and GIS. Addison-Wesley Longman Publishing Co., Inc., Boston, 1990.
4. Tzouramanis, T., Vassilakopoulos, M., Manolopoulos, Y.: Overlapping Linear Quadtrees: a Spatio-temporal Access Method. Proc. 6th ACM Symposium on Advances in Geographic Information Systems (ACM-GIS'98), 1998, pp. 1–7.

Special Task Scheduling and Control of Cluster Parallel Computing for High-Performance Ground Processing System*

Wanjun Zhang^{1,2}, Dingsheng Liu¹, Guoqing Li¹, and Wenyi Zhang¹

¹ Key laboratory, China Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, No 45 Bei San Huan Xi Road, Beijing, 100086, China
{zhangwanj, dsliu, gqli, wyzhang}@ne.rsgs.ac.cn

² Graduate University of the Chinese Academy of Sciences

Abstract. This paper mainly discusses the special problems and solutions for multi-task and data flow control in cluster parallel computing system which dedicated used for High-performance Remote Sensing Satellite Ground Pre-processing System (GHIPS). After giving the overview of the GHIPS, the structure and function of Operation and Mission Subsystem (OMS) shall be formulated. The more detail discussion shall be focused on the organization and processing mechanism based on workflow of the task procedure as well as the task scheduling strategies. Based on our experiences more flexible and reasonable solutions will be given.

1 Introduction and System Background

Satellite ground processing system is used to process satellite remote sensing data and produce standard image products for application. Since the huge of downlink data from satellite and the complex computation of algorithms, cluster parallel computing appeared as a promising way^{[1][2]}. With the development of spacecraft technology more and more huge data are acquired from satellite and the requirements on the performance of ground processing system are increasing dramatically.

The High-performance Remote Sensing Satellite Ground Pre-processing System (GHIPS), which is the first general ground pre-processing system based on cluster parallel computing, was successfully developed by our team and can be regarded as a multi-purpose, multi-user and multi-product processing system. In particular, the system was integrated into parallel and high-performance computational environment to provide powerful processing ability. Now it has been successfully operated as an operational processing system for BEIJING-1, which has been successfully launched on 27 Oct. 2005. Most notably, the design and architecture of GHIPS are not only support BEIJING-1, but also can apply to other satellites data with minor modifications.

Each task in GHIPS will be processed by a serial of steps, including cloud evaluation, systemic radiometric and geometric correction, and so on. It was appeared to be crucial for the whole system how to organize such steps to form reasonable processing workflows

* The work was supported by the National Key Science and Technology Research Program, Ministry of Science and Technology, China.

and how to control the running of such workflows with multi-task scheduling in the cluster parallel computing ground processing system. In GHIPS, such functions was implemented and integrated as the Operation and Mission Subsystem (OMS). This paper will describe the structure and characteristics of OMS, and then give some architecture analysis and implementation introduction. Further issues will focus on special problems related to task scheduling and data flow control in parallel environment, and then some preliminary solutions as well as its test results shall be given.

2 Functions and Structure of OMS

2.1 OMS Functions

OMS serves as the task scheduling and central control module. Nearly all of the other subsystems of GHIPS, including Data Archive Subsystem, Radiation Calibration, Geometric Calibration, Value-added Processing, etc, will interact with it.

OMS is divided into two parts: one is called OMS Master (OMS-M) running at cluster side to schedule and control the tasks, and the other is used to interact with users, which could be called OMS Client running at client workstation and provide simply operational interfaces for users to manipulate the clusters.

The detail mission of the OMS is including:

1. User Interface Management

OMS provides a friendly interactive user interface, which can be used for high speed downlink raw data separation, cloud evaluation, geometric correction with GCPs, image fusion, processing parameter setting, system configuration, and processing progress controlling etc. It can encapsulate all the parameter and control information to Object Description Language (ODL) ^[3] scripts, which can be parsed and executed by server-side daemon service of OMS-M. The status of each processing step and result can also be monitored in the interface.

2. Task Scheduling and Process Control for Clusters:

It is one of the core components of the OMS. When a task is submitted, OMS will schedule and allocate resources in clusters to process. The task processing flow can be controlled by the operators' commands coming from interfaces. Operators can cancel, hold or suspend the tasks processing of tasks when necessary. So the flexible organization of tasks with diverse algorithms, data flow and scale and reliable control of processes in the high-performance system is important issues to be considered.

3. User Authority Management:

There are three level user groups with different rights in OMS, including roots, engineers, and operators. OMS will ensure the security and right for each level user. OMS should manage all the user accounts for operation systems, database, NAS, and each subsystem of GHIPS. It has provided a single-sign-on mechanism to manage all the user profiles.

4. Others:

Many other aspects are involved in OMS. For example, it will monitor the status of all hardware and devices in the system, generate statistical reports of tasks for analysis, and manage log events for analysis etc.

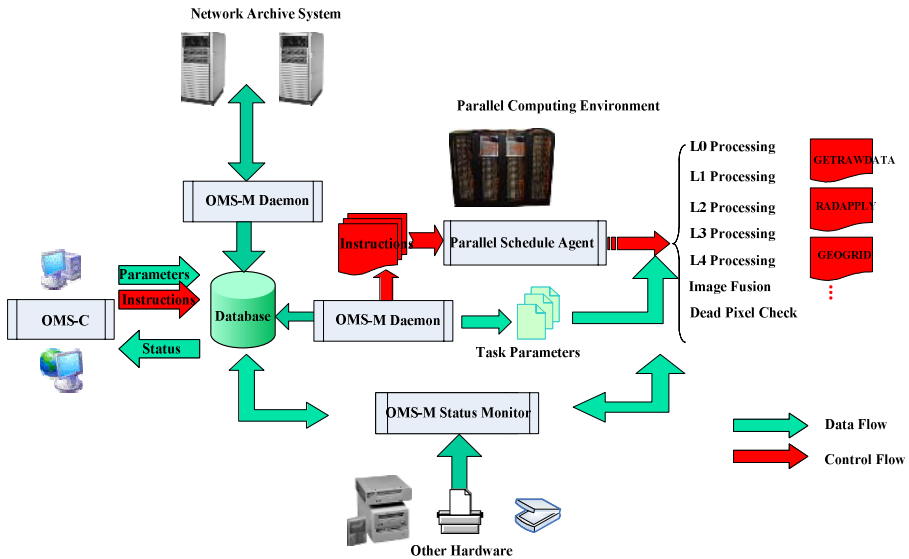


Fig. 1. Structure of Operation and Mission Subsystem

2.2 OMS Structure

The structure and organization of OMS can be showed in figure-1 based on the above discussion.

The cluster-based parallel computation environment provides near real-time processing of massive downlink data. The product generating function can provides standard data products as well as some value-added products. Meanwhile, the NAS data archive and the comprehensive user interface plus the above modules form this multi-platform system. And besides, varieties of I/O hardware link to the system for data transmission and storage, such as scanners and super-DLT tape library. All these factors make the management related to task schedule and data flow control much complex and crucial in high-performance environment.

3 Crucial Problems and Solutions

3.1 Management of Processing Flow

The flexible organization of multi-task with diverse algorithms, data flow and scale in the high-performance system is the first important issue to be considered. When a processing flow is performed, it should deal with the different satellite parameters, select appropriate radiation calibration algorithms and modes, and organize the processing sequence. Workflow^[4] related concepts have been applied to the design and implementation for the processing of tasks in OMS.

As shown in Figure 1, a serial of process activities and prototype steps, such as GETRAWDATA, RADAPPLY, GEOGRID, and RESAMPLE... etc., which can be

reused in many typical processing flows, and each flow has its distinctive parameters, I/O interfaces, special algorithms and many other aspects. For example, the standard Level-3 and Level-4 products generating can be achieved with different serials. Object Define Language (ODL) describes the processing flow using object-orient design pattern with defined schema. Each ODL file contains the process definition and activity, transition information, participant declaration, and organization model.

The following script is typical ODL containing task parameter and control information example.

Server-side OMS parser can be seen as workflow engine. It will load and translate the scripts into binary codes for clusters. It can get the transition information dynamically and organize the processing activity sequence with relevant modules and parameters. Thus tasks with new algorithms and parameters will be easily organized by modifying the item value of the script, not changing programs. The Process flow can be arranged at design time expressed by ODL definition script. Because the system is in high-performance and parallel environment, there are some processing activities should run by parallel mode. OMS can dynamically determine the mode of steps with correspondent algorithms and implementation when perform parse the process definition. For example, when doing resample, OMS can schedule the operation in one node with serial mode, and it can also be performed in all nodes with Parallel Remote Sensing Image Processing File System (PIPFS) [5].

```

OBJECT                = CONTROL
    JOBID              = DMC4_00000858
    COMMAND            = "S"
    LEVEL_INPUT        = 0
    LEVEL_OUTPUT       = 4
    NEWJOB             = "Y"
END_OBJECT            = CONTROL
OBJECT                = RADAPPLY
    LEVEL              = 1
    L1_SKIP            = "N"
    PARALLEL           = "Y"
    PRODFORMAT         = 1
    L1_BITSPERPIXEL   = 8
    STRRADPARM         = 11
    RADCAL_DIR         = ""
    DISPOSE_DEADPIXEL = "N"
END_OBJECT            = RADAPPLY
OBJECT                = GEOGRID
    LEVEL              = 2
    STRGEOPARM         = 21
    GEOCALFILE         = ""
    FRAME_TYPE         = 1
    GRIDSIZE           = {32, 32}
    RESAMPLE           = "CC"
    PROJCODE           = "+proj=tmerc +k
                        =1 +x_0=500000 +y_0=0 "
    SPHEROID           = "krass"
    PIXELSIZE          = {32.00, 32.00}
END_OBJECT            = GEOGRID

```

Operators could modify the processing status after the task is scheduled and executed. The commands include cancel, hold, suspend or restart a specified task processing. The dynamical transformation of the tasks status can be shown in figure 2.

In order to response the command and status converting, OMS-M sets checkpoints on each activity of the flow. When the activity is finished, it will check the execution logs, the current status and the user commands. Then the process management mechanisms of Linux operation system will be called by OMS to control the status change and the results could be passed to operators immediately. In addition, the interface of OMS enables operators to control the flow of both processing and data automatically or manually.

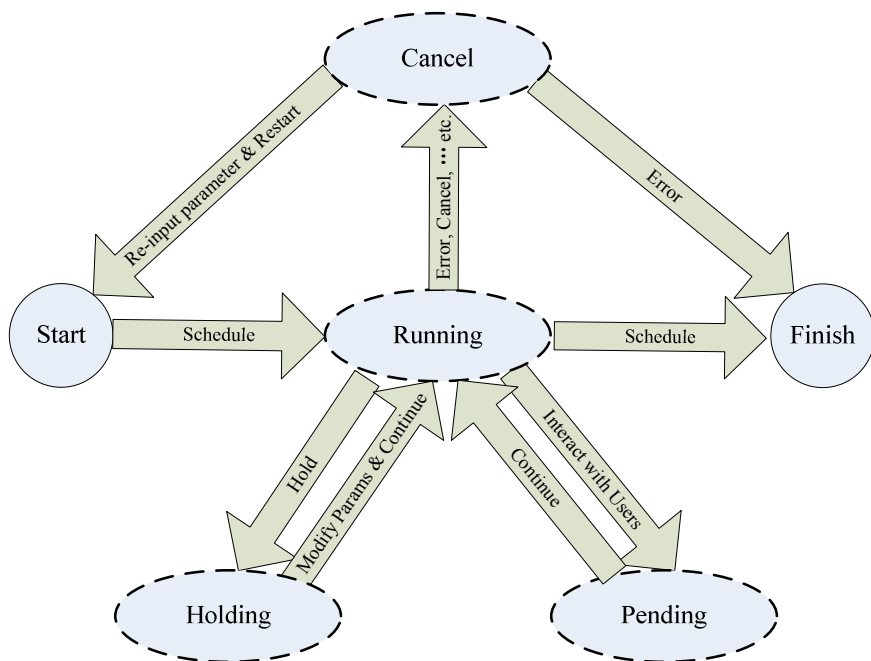


Fig. 2 Tasks Status Transformation in OMS

3.2 Schedule Strategies

The efficient control of the processes and data flow in clusters is also essential for the system, which can be seen from the figure 1. Although there are many perfect task management tools in Linux Cluster platform, OMS cannot use them directly. The main problem here is how to control the process between clusters and user interface clients. As mentioned in the last chapter, OMS has two parts: OMS-M and OMS-C. The OMS-C created preliminary ODL files which include task instructions and parameters based on user input, and the ODL file is swapped in a schedule pool shared by the OMS-M. The OMS-M can be viewed as an agent which transfers ODL instructions and parameters to cluster tasks and then those tasks can be performed by task

management tools. As a response, the agent will adjust the parallel resource to meet the needs of OMS. All the progress and status of performance will return to OMS-M, and it can pass the message to schedule pool, by which the OMS-C will acquire the run time information. The parallel intra-mechanism of such agents in clusters is similar with common tools and also can be modified from the latest update of such tools, which is separated from the OMS. With the mechanism, OMS will not be affected by the hardware update of clusters.

The system was architected based on Parallel Remote Sensing Image Processing File System (PIPFS), which can support operating massive data parallelized in clusters environment. Thus OMS has two categories parallel schedule. One is task-level and it mainly exerts for none computation-sensitive operations, such as get raw data, put product, etc. The other is algorithm-level. It will be used for the computation-sensitive algorithms and other massive data processing based on MPI and PIPFS, such as resample, rotation, and calibration. The parallel schedule strategy can be defined at design time, or changed at running time with modified the flag of processing parameter in ODL. These two strategies ensure the real-time capability of processing multi-task with massive data.

There are many strategies involved in task management. The topological architecture of the clusters is designed as multi NFS nodes to distribute the data efficiently, which is the key issue for massive remote sensing image processing. All the computing resources are allocated for tasks with different priority based on a FIFO queue, and the queue can support to schedule and process a large number of tasks simultaneously. The parallel schedule agent on clusters will assign the top priority task to the node with maximum idle resources. As a result, load-balance can be achieved dynamically.

Database is widely used in OMS, including OMS-C and OMS-M side. GHIPS is architected on heterogeneous platforms, such as the Microsoft Windows operations system of OMS-C, the clusters on Redhat Linux, and Network Archive System. At the same time there are many hardware and devices, including DVD-recorders, tapes, printer and scanner, to be controlled. Database is a good solution to communicate and exchange information between platforms, and monitor the status of devices. OMS has special services based on Oracle9i to serve as proxy and stub among the heterogeneous platforms for interchanging status. High speed database access engines which include Microsoft ADO^[6] and Oracle OCCI^[7] are used in the implementation of the services.

4 Results and Future Work

Through the successful integrating these solutions in GHIPS, a good performance was obtained. OMS can generate nearly fifteen processing flows and new processing will be easily added to the system. There are three radiation and geometric calibration modes managed by OMS, and each mode has different algorithms. End-users can select more than 20 projections to generate products. In the test-bed, OMS can stably schedule about 100 tasks with 30MB/s through-out capability simultaneously. At the same time, OMS can deal with the MS and infrared data of CBERS-1 by changing satellite parameters in OMS.

Based on the current work, the next step about OMS will focus on the flexible design and runtime control for processing using graphic tools. The standard of Ground Station Processing Markup Language, which is based on XML and consistent with WFMC schema^[8] of Workflow Management Coalition, is putting forward. Many other relevant design toolkits are developing, including computational resource organization chart, processing flow designer studio, and automation parser. Our efforts will provide a state-of-the-art design for management of schedule and control in ground processing system based on parallel and cluster computation environment.

References

1. Moon-Gyu Kim, Sung-Og Park, Ji Hyeon, Sung-Og Park etc, Development of Satellite Image Ground Receiving and Processing System for High Resolution Satellites, <http://www.gisdevelopment.net/aars/acrs/2002/vhr/103.pdf>
2. Chao-Tung Yang, Chih-Li Chang, Using a Beowulf Cluster for a Remote Sensing Application, in 22nd Asian Conference on Remote Sensing, Nov.2001
3. NASA, Object Description Language Specification and Usage, <http://pds.jpl.nasa.gov/documents/sr/Chapter12.pdf>
4. Van der Aalst, W., M., P. , Barros, A., P., ter Hofstede, etc, Advanced Workflow Patterns, in Conference on Cooperative Information Systems, pp. 18–29, 2000.
5. Zhu Yaofei, Li Guoqing, The research and experimentation of parallel file system in remote sensing image parallel processing system, Master's thesis, China remote sensing satellite ground station.
6. Shi Jun, Ge Jun, Programming ADO, Tsinghua University Press, 2001
7. Oracle, Oracle C++ Call Interface Programmer's Guide, <http://otn.oracle.com/>
8. Workflow Management Coalition, <http://www.wfmc.org/>

AMEEPAR: Parallel Morphological Algorithm for Hyperspectral Image Classification on Heterogeneous Networks of Workstations

Antonio Plaza, Javier Plaza, and David Valencia

Department of Computer Science, University of Extremadura
Avda. de la Universidad s/n, E-10071 Caceres, Spain
{aplaza, jplaza, davaleco}@unex.es

Abstract. Hyperspectral imaging is a new technique in remote sensing that generates hundreds of images corresponding to different wavelength channels for the same area on the surface of the Earth. Most available techniques for hyperspectral image classification focus on analyzing the data without incorporating the spatial information; i.e. the data is treated not as an image but as an unordered listing of spectral measurements where the spatial coordinates can be shuffled arbitrarily without affecting the final analysis. Despite the growing interest in the development of techniques for interpretation and classification of such high-dimensional imagery, only a few efforts devoted to the design of parallel implementations exist in the open literature. In this paper, we describe AMEEPAR, a parallel morphological algorithm that integrates the spatial and spectral information. The algorithm has been specifically optimized in this work for execution on heterogeneous networks of workstations. The parallel properties and classification accuracy of the proposed approach are evaluated using four networks of workstations distributed among different locations, and a massively parallel Beowulf cluster at NASA's Goddard Space Flight Center.

1 Introduction

The rapid development of space and computer technologies has made possible to store a sheer volume of remotely sensed image data, collected from heterogeneous sources. In particular, NASA is continuously gathering imagery data with hyperspectral Earth observing sensors such as Jet Propulsion Laboratory's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)¹, which covers the wavelength region from 0.4 to 2.5 μm using 224 spectral channels at nominal spectral resolution of 10 nm (see Fig. 1). The incorporation of hyperspectral sensors on airborne/satellite platforms is currently producing a nearly continual stream of high spatial and spectral resolution data, and this high data volume demands efficient and robust data analysis techniques.

The underlying assumption governing most available techniques for hyperspectral analysis is that each pixel vector measures the response of multiple underlying materials at each site. A hyperspectral image (sometimes referred to as "image cube") is often a combination of two situations: a few sites in a scene are pure macroscopic

materials, e.g., soil or water, but many other are mixtures of materials. For instance, the vegetation pixel in Fig. 1 may comprise a mixture of different types of vegetation, soil, atmospheric interferers, etc. Further, most available techniques only use the spectral information available in the image data. Therefore, such techniques would yield the same result for a data cube, and for the same data cube where the spatial positions have been randomly permuted. By taking into account the complementary nature of spatial and spectral information in simultaneous fashion, it is possible to alleviate the problems related to each of them taken separately.

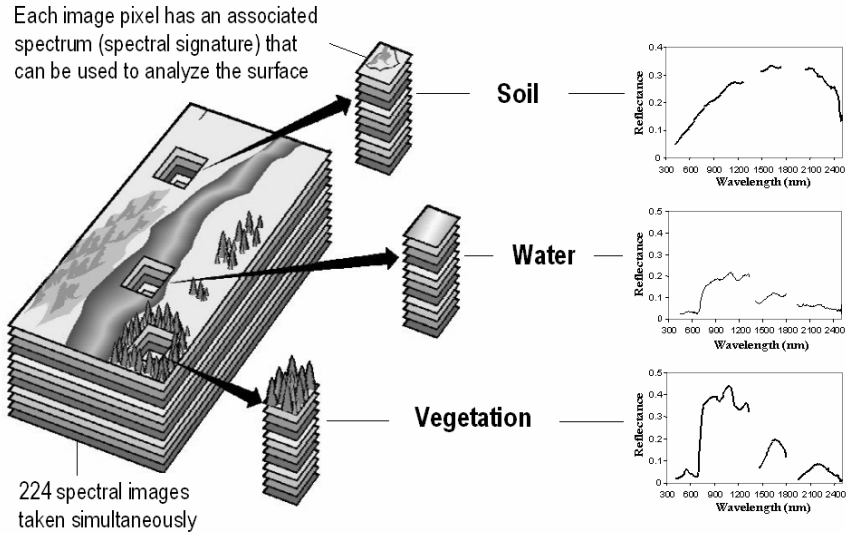


Fig. 1. Hyperspectral imaging. Each pixel is given by a vector of values or “spectral signature”.

While integrated spatial/spectral developments hold great promise, they also introduce new processing challenges. In turn, many applications require a quick response (e.g., target detection for homeland defense/security purposes, risk/hazard prevention/response including wild land fire tracking, biological threat detection, monitoring of oil spills and other types of chemical contamination). In recent years, several efforts have been focused on the incorporation of high-performance computing (HPC) models in remote sensing applications. Unfortunately, only a few research efforts devoted to HPC-based hyperspectral imaging exist in the open literature (which is partly due to non-disclosure restrictions in some cases). This paper develops an efficient parallel algorithm for spatial/spectral analysis in heterogeneous computing environments², which are expected to become a tool of choice in many on-going and planned remote sensing missions. The method is a parallel version of the Automated Morphological Endmember Extraction (AMEE) algorithm³, which integrates the spatial and spectral information in the data. The paper is organized as follows. In the following section, we introduce the AMEE algorithm and its parallel implementation. The parallel algorithm is then evaluated using four networks of workstations and a Beowulf cluster at NASA’s Goddard Space Flight Center. The paper concludes with some remarks.

2 Parallel Algorithm for Hyperspectral Image Classification

The AMEEMPAR algorithm is based on mathematical morphology³, a classic nonlinear spatial processing technique that provides a remarkable framework to achieve the desired integration of spatial and spectral responses. First, we provide an overview of the AMEE algorithm. Then, we provide a description of its parallel implementation.

2.1 AMEE Algorithm

The AMEE algorithm relies on two basic morphological operations: erosion and dilation³. Let us denote by $f(x, y)$ the pixel vector at spatial coordinates (x, y) of a hyperspectral scene. The erosion of f by B (a so-called “structuring element”) consists of selecting the “minimum” pixel vector in the spatial neighborhood of $f(x, y)$ defined by B . Similarly, the dilation of f by B consists of selecting the “maximum” pixel vector (called endmember in hyperspectral analysis terminology). We provide below a version of AMEE which is tuned for unsupervised classification of hyperspectral data.

AMEE algorithm

Inputs: Image cube f , B , Number of iterations I_{max} , Number of endmembers p .

Output: MEI image.

1. Set $i=1$ and initialize $MEI(x, y) = 0$ for each pixel.
2. Move B through all the pixels of f , defining a local spatial search area around each $f(x, y)$, and calculate the maximum and the minimum pixel at each B -neighborhood using morphological dilation and erosion, respectively, as follows:

$$(f \oplus B)(x, y) = \{f(x-s', y-t'), (s', t') = \arg \max_{(s,t) \in Z^2(B)} \{D_B[f(x-s, y-t)]\}$$

$$(f \ominus B)(x, y) = \{f(x+s', y+t'), (s', t') = \arg \min_{(s,t) \in Z^2(B)} \{D_B[f(x+s, y+t)]\}$$

3. Update $MEI(x, y)$ at each pixel using $MEI(x, y) = SAD[(f \oplus B)(x, y), (f \ominus B)(x, y)]$, where SAD is the spectral angle distance³, i.e., the arc cosine of the vector dot product divided by the product of the norms.
4. Set $i = i + 1$. If $i = I_{max}$ then go to step 4. Otherwise, set $f = f \oplus B$ and go to 2.
5. Select the set of p pixel vectors in f with higher associated score in the resulting MEI image (called endmember pixels) and form a unique spectral set of $q \leq p$ pixels by calculating the SAD for all vector pairs. Estimate the fractional abundance, $\alpha_i(x, y)$, of those q signatures at $f(x, y)$ using a linear mixture model.
6. Obtain a classification label for each pixel $f(x, y)$ by assigning it to a class given by the endmember with the highest fractional abundance score in that pixel. All estimated abundance fractions $\{\alpha_1(x, y), \alpha_2(x, y), \dots, \alpha_q(x, y)\}$ are compared, and the one with the maximum value is found, say:

$$\alpha_{i^*}(x, y), \text{ with } i^* = \arg \left\{ \max_{1 \leq i \leq q} \{\alpha_i(x, y)\} \right\}.$$

2.2 AMEEPAR Algorithm

A major requirement for efficient parallel algorithms on distributed memory systems is finding a data decomposition that minimizes the communication between the processors⁴. For that purpose, AMEEPAR adopts a spatial-domain partitioning approach, in which the same pixel vector is never split among several processors (in spectral-domain parallel, the structuring element-based calculations made for each hyperspectral pixel need to originate from several processing elements, and thus require intensive inter-processor communication). A second important issue in the design of the AMEEPAR is that redundant information (*overlap* borders) are added to local partitions to avoid accesses outside the partition domain when the structuring element computation requires pixel vectors from other partitions⁵.

The algorithm has been implemented in the C++ programming language using calls to message passing interface (MPI). It uses a master-slave paradigm in which the master scatters hyperspectral data without creating partial data structures at the root. For that purpose, we make use of MPI *derived datatypes* to directly scatter data structures, which may be stored non-contiguously in memory, in a single communication step. In order to slice the available data into chunks, there is a need to balance the workloads of k heterogeneous resources so that each processor P_i will accomplish a

share α_i of the total workload, with $\alpha_i \geq 0$ for $1 \leq i \leq k$ and $\sum_{i=1}^k \alpha_i = 1$.

AMEEPAR algorithm

Inputs: Image cube f , B , Number of iterations I_{max} , Number of endmembers p .

Output: MEI image.

1. Obtain necessary information about the heterogeneous system, including the number of available processors, k , each processor's identification number, $\{P_i\}_{i=1}^k$, and processor cycle-times, $\{w_i\}_{i=1}^k$.
2. Determine the total volume of information, R , that needs to be replicated from the original data volume, V , in accordance with the adopted border overlap strategy.
3. Let the total workload to be handled by the algorithm be given by $W = V + R$.
4. Set $\alpha_i = \left\lfloor \frac{(p/w_i)}{\sum_{i=1}^k (1/w_i)} \right\rfloor$ for all $i \in \{1, \dots, k\}$.
5. For $m = \sum_{i=1}^k \alpha_i$ to $(V + R)$ find $j \in \{1, \dots, k\}$, so that $w_j \cdot (\alpha_j + 1) = \min\{w_i \cdot (\alpha_i + 1)\}_{i=1}^k$ and set $\alpha_k = \alpha_k + 1$.
6. Use the resulting $\{\alpha_i\}_{i=1}^k$ to obtain a set of k spatial-domain heterogeneous partitions of f , and send its corresponding partition to each processor along with B .
7. Broadcast B , I_{max} , p to heterogeneous processors, and execute AMEE in parallel.
8. Collect all the individual classification results provided by each processor P_i , and merge them together to form a final classification image.

It should be noted that a homogeneous version of the AMEEPAR algorithm above (called HomoAMEEPAR)⁵ can be obtained by rewriting step 4 as $\alpha_i = p/w_i$ for all $i \in \{1, \dots, k\}$, where w_i is a constant communication speed between each processor pair.

3 Experimental Results

Before describing our results, we introduce the parallel computing architectures used in experiments, which include four networks of workstations and a Beowulf cluster:

1. *Fully heterogeneous network.* Consists of 16 different SGI, Solaris and Linux workstations, and four communication segments. Table 1 shows the cycle-times of the processors. The communication network consists of four communication segments interconnected by three slower communication links with capacities $c^{(1,2)} = 29.05$, $c^{(2,3)} = 48.31$, $c^{(3,4)} = 58.14$ in milliseconds. Table 2 shows the capacity of all point-to-point communications, expressed as the time in milliseconds to transfer a one-megabit message between each pair (P_i, P_j) in the network.
2. *Fully homogeneous network.* Consists of 16 identical Linux workstations with processor cycle-time of $w = 0.0131$ seconds per megaflop, interconnected via a homogeneous communication network with capacity $c = 26.64$ milliseconds.
3. *Partially heterogeneous network.* Formed by the set of 16 heterogeneous workstations in Table 1 but interconnected using the same homogeneous communication network with capacity $c = 26.64$ milliseconds.
4. *Partially homogeneous network.* Formed by 16 identical Linux workstations with processor cycle-time of $w = 0.0131$ seconds per megaflop but interconnected using the heterogeneous communication network shown in Table 2.
5. *Thunderhead Beowulf cluster.* Formed by 256 identical 2.4 GHz Intel Xeon nodes, each with 1 GB of memory and 80 GB of main memory (see <http://newton.gsf.nasa.gov/thunderhead>).

Table 1. Processor cycle-times (in seconds per megaflop) for the heterogeneous processors

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}
.005	.010	.020	.007	.010	.007	.007	.010	.007	.045	.013	.013	.013	.013	.013	.013

According to a recent study², the four networks of workstations above can be considered *equivalent* since they satisfy the following three principles: 1) they have the same number of processors; 2) the average processor speed is the same in all cases; and 3) the aggregate characteristics of the communication network are all the same. At this point, we reiterate that the configuration of the four networks above was carefully designed to make sure that the three principles above were satisfied, and the aggregate performance was the same.

The AMEEPAR algorithm (and its homogeneous version) were applied to a hyper-spectral scene collected by the AVIRIS sensor, which consists of 2048x614 pixels, 224 spectral bands, and moderate spatial resolution (20-meter pixels). It was gathered over the Indian Pines region in Indiana, and represents a challenging classification

problem. Part of these 275 MB data set are available online, along with ground-truth, from <http://dynamo.ecn.purdue.edu/~biehl/MultiSpec>. In experiments, the structuring element size was fixed to $B_{3 \times 3}$ to reduce the amount of redundant computations³. The number of iterations I_{max} was increased from 1 to 5 (according to our implementation, increasing the value of I_{max} is equivalent to considering a larger spatial context). Finally, we set the number of endmembers to be extracted, p , to 16 after estimating the intrinsic dimensionality of the data¹.

Table 2. Capacity of links (measured by the time in milliseconds to transfer a one-megabit message) for the heterogeneous communication network

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}
P_1	-	19.2	19.2	19.2	48.3	48.3	48.3	48.3	96.6	96.6	154	154	154	154	154	154
P_2	19.2	-	19.2	19.2	48.3	48.3	48.3	48.3	96.6	96.6	154	154	154	154	154	154
P_3	19.2	19.2	-	19.2	48.3	48.3	48.3	48.3	96.6	96.6	154	154	154	154	154	154
P_4	19.2	19.2	19.2	-	48.3	48.3	48.3	48.3	96.6	96.6	154	154	154	154	154	154
P_5	48.3	48.3	48.3	48.3	-	17.6	17.6	17.6	48.3	48.3	106	106	106	106	106	106
P_6	48.3	48.3	48.3	48.3	17.6	-	17.6	17.6	48.3	48.3	106	106	106	106	106	106
P_7	48.3	48.3	48.3	48.3	17.6	17.6	-	17.6	48.3	48.3	106	106	106	106	106	106
P_8	48.3	48.3	48.3	48.3	17.6	17.6	17.6	-	48.3	48.3	106	106	106	106	106	106
P_9	96.6	96.6	96.6	96.6	48.3	48.3	48.3	48.3	-	16.3	58.1	58.1	58.1	58.1	58.1	58.1
P_{10}	96.6	96.6	96.6	96.6	48.3	48.3	48.3	48.3	16.3	-	58.1	58.1	58.1	58.1	58.1	58.1
P_{11}	154	154	154	154	106	106	106	106	58.1	58.1	-	14.2	14.2	14.2	14.2	14.2
P_{12}	154	154	154	154	106	106	106	106	58.1	58.1	14.2	-	14.2	14.2	14.2	14.2
P_{13}	154	154	154	154	106	106	106	106	58.1	58.1	14.2	14.2	-	14.2	14.2	14.2
P_{14}	154	154	154	154	106	106	106	106	58.1	58.1	14.2	14.2	14.2	-	14.2	14.2
P_{15}	154	154	154	154	106	106	106	106	58.1	58.1	14.2	14.2	14.2	14.2	-	14.2
P_{16}	154	154	154	154	106	106	106	106	58.1	58.1	14.2	14.2	14.2	14.2	14.2	-

Table 3. Execution times and classification accuracies on the four networks of workstations

Algorithm	AMEEPAR			HomoAMEEPAR		
	1	3	5	1	3	5
Fully heterogeneous network	284	321	379	1456	1528	1589
Fully homogeneous network	298	339	372	281	317	355
Partially heterogeneous network	288	325	365	1223	1267	1301
Partially homogeneous network	294	336	371	437	472	512
Classification accuracy (%)	69.23	74.48	91.05	69.23	74.48	91.05

Table 3 shows the classification accuracies and execution times obtained by both algorithms in the four considered networks of workstations. As expected, the execution times reported on Table 3 show that the heterogeneous algorithm was able to adapt much better to fully (or partially) heterogeneous environments than the homogeneous version, which only performed satisfactorily on the fully homogeneous network. One can see that AMEEPAR was several times faster than its homogeneous counterpart in the fully heterogeneous network, and also in both the partially

homogeneous and the partially heterogeneous networks. On the other hand, the HomoAMEEPAR algorithm only slightly outperformed its heterogeneous counterpart in the fully homogeneous network. Table 3 also reveals that the performance of the heterogeneous algorithm on the fully heterogeneous network was almost the same as that evidenced by the homogeneous algorithm on the fully homogeneous network. This reveals that the heterogeneous algorithm was very close to the optimal heterogeneous modification of the basic homogeneous one².

In order to explore load balance of the two algorithms above on the four networks of workstations, Table 4 shows the imbalance scores achieved by the different algorithms (implemented with I_{max} set to 5 iterations). The imbalance is defined as $D = R_{max} / R_{min}$, where R_{max} and R_{min} are the maxima and minima processor run times, respectively. Therefore, perfect balance is achieved when $D=1$. In the table, we report the imbalance considering all processors, D_{All} , and also considering all processors but the root, D_{Minus} . In all cases, load balance was similar when the root processor was not included, which means that the master node does not have high computation load. It is also clear from Table 4 that the homogeneous algorithm executed on the heterogeneous network provided the highest values of D_{All} and D_{Minus} (and hence the highest imbalance), while the heterogeneous algorithm always resulted in values of D_{All} and D_{Minus} which were closer to 1, regardless of the platform where it was run.

Table 4. Load balancing rates on the four networks of workstations

Algorithm	AMEEPAR		HomoAMEEPAR	
	D_{All}	D_{Minus}	D_{All}	D_{Minus}
Fully heterogeneous network	1.09	1.02	1.51	1.47
Fully homogeneous network	1.16	1.07	1.08	1.02
Partially heterogeneous network	1.11	1.03	1.44	1.41
Partially homogeneous network	1.13	1.05	1.33	1.26

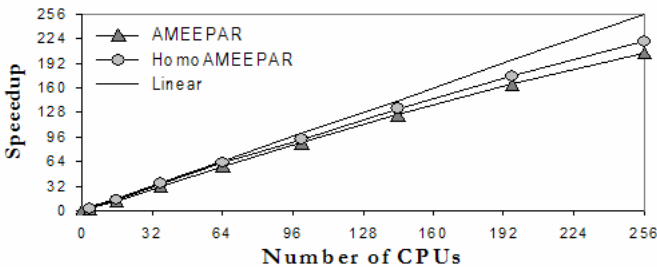


Fig. 2. Scalability of AMEEPAR and HomoAMEEPAR on the Thunderhead Beowulf cluster

Finally, and with the ultimate goal of exploring issues of scalability and portability of heterogeneous algorithms to existing massively parallel computing platforms (which are mainly homogeneous in nature), we have also compared the performance of the

two algorithms on the Thunderhead Beowulf cluster. Fig. 2 shows the speedups achieved by AMEEPAR (and its homogeneous version) over a single-processor run of the sequential AMEE algorithm on Thunderhead, all of them implemented with I_{max} set to 5. As Fig. 2 shows, the scalability of AMEEPAR was similar to that achieved by its homogeneous prototype on the Beowulf cluster. Although AMEEPAR introduces redundant calculations, which are expected to slow down the computation *a priori*, the measured speedups tend to be higher for large structuring element sizes, a fact that reveals that the proposed scheme scales better as the size of the problem increases. As a result, the dominant issue in AMEEPAR is problem size, which makes the algorithm particularly appealing for high-dimensional imaging applications. To conclude this section, we must also note that AMEEPAR was able to provide near real-time classification performance in the Thunderhead cluster. The algorithm only required 8 seconds to produce a result using 256 processors, and 30 seconds with 64 processors. This is a relevant achievement, in particular, if we take into account that 1874 seconds were required to process the entire data set using a single Thunderhead processor.

4 Conclusions

This paper provided an investigation of a parallel morphological technique to extract relevant information from hyperspectral image data sets in heterogeneous computing environments. Experimental results reveal that the proposed algorithm offers a simple, yet highly scalable and relatively platform-independent solution in the context of hyperspectral image classification applications. Although many available approaches do not take into account the spatial information explicitly (a fact that has been perceived as an advantage for the development of parallel implementations), experimental results in this paper suggest that spatial/spectral classification approaches may indeed be tuned for “pleasingly parallel execution” due to the windowing nature of such algorithms, and also because they can effectively balance the load in heterogeneous systems. The proposed method seems ideally suitable for data mining applications, which previously looked too computationally intensive due to the immense data archives common to remote sensing problems. Combining the readily available computational power offered by heterogeneous platforms with the new sensor instruments may introduce major changes in the systems used for exploiting Earth and planetary data.

References

1. Green, R.O., et al.: Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). *Remote Sensing of Environment*, 65 (1998) 227–248
2. Lastovetsky, A., Reddy, R.: On Performance Analysis of Heterogeneous Parallel Algorithms. *Parallel Computing*, 30 (2004) 1195–1216
3. Plaza, A., Martinez, P., Perez, R., Plaza, J.: Spatial/Spectral Endmember Extraction by Multidimensional Morphological Operations. *IEEE Transactions on Geoscience and Remote Sensing*, 9 (2002) 2025–2041
4. Seinstra, F.J., Koelma, D., Geusebroek, J.M.: A Software Architecture for User Transparent Parallel Image Processing. *Parallel Computing* 28 (2002) 967–993
5. Plaza, A., Valencia, D., Plaza, J.: Commodity Cluster-Based Parallel Processing of Hyperspectral Imagery. *Journal of Parallel and Distributed Computing* 66 (2006) 345–358

Visual Discovery and Reconstruction of the Climatic Conditions of the Past*

Roberto Therón

Departamento de Informática y Automática,
Universidad de Salamanca, Salamanca, 37008, Spain
theron@usal.es

Abstract. The development of new tools and methodologies is necessary in order to better understand current and past climatic changes. To be useful, these mathematical or software tools must not remain only in the hands of specialists in statistics, but must also be usable by the larger community of paleoclimatologists. It is therefore necessary to conceive a user interface adapted to the specificities of their use in paleoclimatology. Here, we propose the development of new tools of interactive analysis. Through the combination of techniques coming from knowledge discovery and information visualization (visual data mining), rapid and accurate paleoclimatic reconstructions will be easier to produce.

1 Introduction

While the need to foresee abrupt climatic changes is an urgent challenge for the society, paleoclimate research has shown that the causes and effects of these changes are very different, with extremely rapid variations even on one-year basis. Computers have played a key role in our understanding of the climatic dynamics. Nowadays, the improvement of data acquisition methods offer us the opportunity to gain the needed depth of information to diagnose and prevent any natural disaster. However, although data are available, the development of new tools and new methodologies is necessary. If very high precision physical or chemical measurements are necessary to reconstruct paleoenvironments, they often need to be accompanied by sophisticated statistical analysis methods ([1],[2]). But, to be useful, these mathematical or software tools must not remain only in the hands of specialists in statistics, but must also be usable by the larger community of paleoclimatologists. It is therefore necessary to foster an optimal use of these mathematical tools, by establishing methodological choices among the most relevant and the most recent statistical methods, and to conceive a user interface adapted to the specificities of their use in paleoclimatology.

The data registered over thousands of years (mainly in ice and sediment cores) is an impressive source of information that, for instance, help us to model earth and oceans dynamics [3], first step to make climatic predictions. When looking for historic climatic data with durations exceeding decades, the largest and

* This work was supported by the MCyT of Spain under Integrated Action (Spain-France) HF2004-0277 and by the Junta de Castilla y León under project SA042/02.

oldest record is found in the oceans. Palaeoceanographers need to manipulate, integrate and analyze time-series that are obtained from a number of independent techniques (such as ocean drilling, ocean tracers, AMC 14C datings, astronomic curves, etc.), which, moreover, are usually produced by different researchers and/or laboratories. This work is done with the aid of proper tools such as PaleoPlot [4] and AnalySeries [5].

Some of these data needed to understand paleoclimate are time-series of specific attributes related to the oceans. Thus, one problem scientists must face is how to know environmental parameters, such as Sea Surface Temperature (SST), at each given past moment. For the reconstruction of this features, isotope measurements ($\delta^{18}\text{O}$) or biomarkers (U_{37}^k index) have been used. On the other hand, for the quantitative reconstruction of environmental conditions of the past, currently the *Modern Analog Technique* (MAT, actually a nearest neighbor prediction)[6], is one of the most commonly used techniques in paleoclimatology.

Although software tools for MAT have been developed [7], and some improvements have arisen such as SIMMAX [8], RAM [2] and artificial neural networks [9], they all have a main drawback: once developed they are black boxes. Paleoclimatologists can use them but no knowledge acquisition is involved; they just trust in the reconstructions obtained, they cannot know if the data used is valid from a geologic point of view. Furthermore, the classic MAT method inherently produces reconstructions whose precision is very difficult to estimate [10].

Visualization provides insight through images and can be considered as a collection of application of specific mappings from the problem domain to a visual range [11]. Thus, it is our aim to design new tailor-made methods of analysing and viewing the paleoclimatic data that would have different advantages and one goal: avoiding blind reconstructions by means of well designed user driven procedures. Through the combination of techniques coming from statistics, information theory, information visualization and visual data mining, rapid and accurate paleoclimatic reconstructions will be easier to produce. In this paper we show how new methods of interactive analysis can be extremely useful for the paleoclimatic field.

The rest of this paper is organised as follows: in section two how modern analogs (neighbors) are found through the calculation of dissimilarity coefficients between modern and paleo data (MAT) is shown. Third section is devoted to explain how a proper interactive analysis enables knowledge discovery and permits more accurate reconstructions. To finalize, the main conclusions and future work are described.

2 Calculating the Distances

This section describes how PaleoAnalogs, a Java based program, improves the modern analog technique in order to provide faster and more accurate reconstructions [12]. It is assumed that the user has faunal census estimates of one or more fossil samples, the core file; and one or more sets of faunal data from modern samples with the related environmental features, the database file. Furthermore,

the user must understand the taxonomic categories represented in the data sets, and be able to recognize taxa that are or may be considered equivalent in the analysis.

The process begins after the selection of the core and database files; in general, these files will contain different taxa (figure 1.a and figure 1.b), both because different taxa are prevalent in different regions and because data providers use varying taxonomic categories (species and subspecies), names, and abbreviations. MAT requires that corresponding variables in different data sets be recognizable as such, otherwise it would be impossible to calculate the distance measures. With the help of the taxa association wizard (figure 1.c) this problem is easily worked out, allowing the user to determine which taxa from both the modern and fossil data files are compared, calculate proportions if needed, and identify the environmental features to be reconstructed.

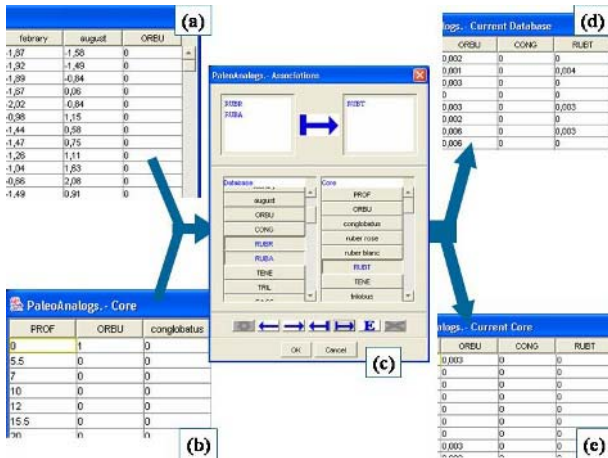


Fig. 1. Taxa association

Once the database and the core data are transformed to have the same number and equivalent taxa (figure 1.d and figure 1.f), each sample in the core is compared with each sample in the database using a dissimilarity coefficient.

Finally, using the distance measure selected by the user, a dissimilarity matrix is built. For each core sample N dissimilarity values are given, being N the number of samples in the modern database; these values are ordered increasingly so that each row of the matrix contains, left-to-right, the list of the N best analogs, that is, the database samples ordered by their alikeness to that particular core pattern.

The next step in MAT is to reconstruct the environmental conditions of each core sample based on the environmental data of a number of best analogs (generally ten). This can be done by calculating the average value or by weighting the analogs. However, this is somehow very strict, because some of the used analogs could not be valid from a geologic point of view and should be eliminated.

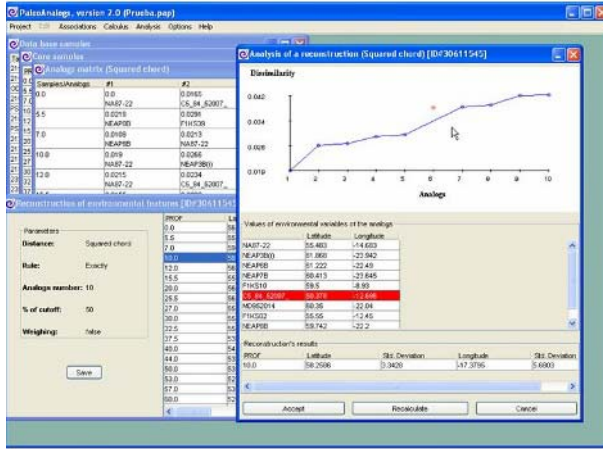


Fig. 2. Reconstruction tuning

It must be noted that the classical MAT technique only permitted to select the number K of analogs (neighbors), normally 10, used to calculate the averaged variables. But consider the case where only 3 of these 10 modern analogs were actually similar to the sample being reconstructed, while the 7 remainders were only the following most similar, and, therefore, it would be a mistake to use them for the reconstruction from a geological point of view.

Figure 2 shows the PaleoAnalog interactive reconstruction tool. It enables for each sample to observe which sites have been closest. For the example in the figure, centimeter 10 of the core (which is a particular year in the past, depending on the sedimentation rate) is being studied. In this case the user has considered that the difference between the dissimilarity values of the fifth and sixth analogs is not acceptable; interactively, by clicking on each point of the graphic the analog is deselected, and the average recalculated using only the remaining selected analogs. On the other hand, the information of each analog is exposed, e.g. sixth analog is site C5-84-S2007, so the paleoclimatologist may decide upon rejecting any particular analog using geological criteria that may suggest not to use the data from that site.

3 Discovering the Past

Although the MAT method is very useful for paleoclimatic reconstruction there is much more information that can be provided than a mere neighbor distance calculation. Thus, before proceed with the algorithmic reconstruction further knowledge can be easily discovered from the calculated set of analogs.

As it has been stated in the previous sections, the problem is that paleontologists can obtain reconstructions as outputs from techniques such as nearest neighbor prediction, but no ways of knowledge acquisition are possible. However,

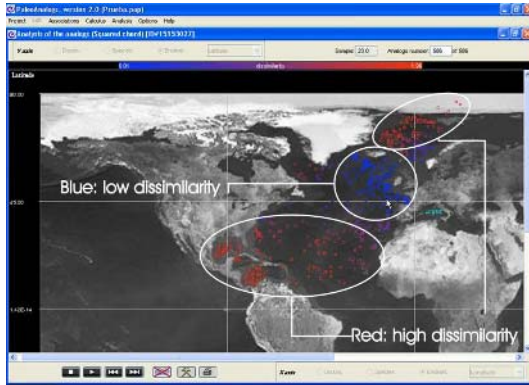


Fig. 3. Analogs geographic distribution

if the particular case of paleoclimatology is considered, *ad hoc* visualization tools may be developed, that will indeed provide insight in that forest of numeric data.

Geologists are trained for geographic visualization, and they basically face a problem of evolution through time, so we can design an interactive visual interface that takes advantage of both location and time.

Let's consider the following situation (figure 3): a paleoclimatologist is studying the data obtained from a particular point in the Mediterranean Sea (the label CORE in the figure shows that point) and the modern data comes from the North Atlantic ocean. Instead of just calculate the temperature reconstruction, he/she can analyse first how the analogs for a given sample (depth/age) are distributed geographically. This can be seen in figure 3, which is the three dimensional (longitude(x axis), latitude(y axis) and dissimilarity(color)) representation for the sample at 20 cm¹. Thus, the expert would easily see(discover) that the studied site, t kiloyears ago had temperatures much more similar to those of cold sites of today (blue zone of analogs in the picture) than those in warm or polar latitudes (red zone of analogs).

Also, this representation can be done for a selected number of analogs (just the number of analogs that will be used for the reconstruction, for instance, or for those with dissimilarity values smaller than a cutoff. Analoges can be labeled with the associated database sample name so the expert might decide that a particular analog is not valid for the reconstruction due to a geological reason.

A combination of visualization approaches may discover a lot of information: choosing to show only 5 analogs, labeling each analog, zooming in and animating the evolution of the whole core, i.e., visualizing the analog evolution through time, we may arrive to some interesting conclusions. Thus, in figure 4 we could start with the deepest sample in the core, i.e., t kiloyears ago: since at that age the planet was covered with ice, we can see that the best 5 analogs are distributed within a wide range of latitudes. As the animation is showing the evolution, we

¹ Depending on the particular age model this depth will be a number of kiloyears in the past.

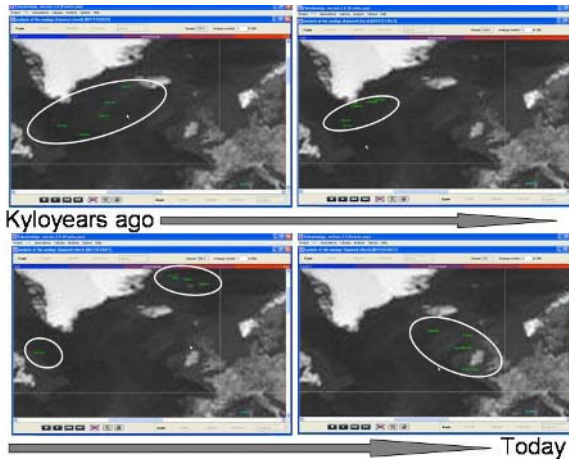


Fig. 4. Best analogs trough time



Fig. 5. Bull Species as an indicator of analog dissimilarity

can see that best analogs are grouped, which is the typical distribution we should expect. The snapshot on the bottom-left shows a particular interesting situation: four of the five analogs are grouped up north (note the blue color), while the fifth one is located at a much warmer latitude (note the red color). This analog distribution should warn the paleoclimatologist, the most probable reason is that the outsider is only the fifth closest neighbor, but not a *real neighbor*, so that particular site should not be considered in the reconstruction for that sample.

Another example (figure 5) of knowledge discovery would be that a particular species (BULL, for instance) is valid as an analog indicator, since the dissimilarity for the sample is very high when the proportion of this species is close to zero (all analogs in the x axis are red and have a high dissimilarity value, while the rest of analogs are getting more and more blue, i.e., more similar, as the proportion of the species grows).

Finally, using specific visual techniques for the interactive analysis of multi-dimensional data such as Parallel Coordinates [11] (see figure 6) will improve the

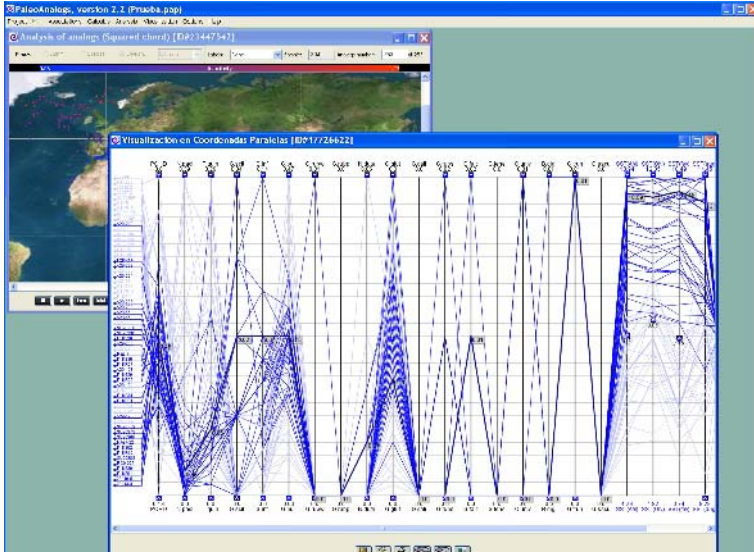


Fig. 6. Reconstruction visually driven by Parallel Coordinates

paleoclimatic knowledge discovery; this practice is an emerging usability issue in geovisualization[13].

4 Conclusions and Future Work

This work is an example of how interactive analysis can help knowledge discovery in the paleoclimatology field. We have shown how a well-known standard technique of the field (MAT) can be greatly improved so the reconstructions of paleoenvironmental conditions can be more accurate. This is accomplished by fostering a user driven reconstruction procedure where the expert get more insight from the data and can decide on the validity of potentials reconstructions. Finally, we can add that more complex interactive analysis can be designed that will help to gain a deeper knowledge about the climatic evolution of a given area.

References

1. Yiou, P., Baert, E., Loutre, M.F.: Spectral analysis of climate data. *Surveys of Geophysics* **17**(6) (1996) 619–663
2. Waelbroeck, C., Labeyrie, L., Deplessy, J., Guoit, J., Labracherie, M., Leclaire, H., Duprat, J.: Improving past sea surface temperature estimates based on planktonic faunas. *Paleoceanography* **13** (1998) 272–283
3. Therón, R., Flores, J.A., Sierro, F.J., Pelejero, C., Grimalt, J., Vaquero, M.: Using data mining and visualization techniques for the reconstruction of ocean paleodynamics. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium. Volume IV.* (2002) 2382–2384

4. Therón, R., Flores, J.A., Sierro, F.J., Vaquero, M., Barbero, F.: Paleoplot: A tool for the analysis, integration and manipulation of time-series paleorecords. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium. Volume VI. (2002) 3528–3530
5. Paillard, D., Labeyrie, L., Yiou, P.: Macintosh program performs time-series analysis. *Eos, Transactions, American Geophysical Union* **77** (1996) 379
6. Hutson, W.H.: The agulhas current during the late pleistocene: Analysis of modern faunal analogs. *Science* **207** (1980) 64–66
7. Schweitzer, P.N.: Analog: A program for estimating paleoclimate parameters using the method of modern analogs. Technical Report 94-645, U. S. Geological Survey Open-File (1994)
8. Pflaumann, U., Duprat, J., Pujol, C., Labeyrie, L.: Simmax: A modern analog technique to deduce atlantic sea surface temperatures from planktonic foraminifera in deep-sea sediments. *Paleoceanography* **11** (1996) 15–35
9. Malmgren, B.A., Kucera, M., Nyber, J., Waelbroeck, C.: Comparison of statistical and artificial neural network techniques for estimating past sea surface temperatures from planktonic foraminifer census data. *Paleoceanography* **16**(5) (2001) 520–530
10. Mannila, H., Toivonen, H., Korhola, A., Olander, H.: Learning, mining or modeling? a case study from paleoecology. In: *Discovery Science*. (1998) 12–24
11. Inselberg, A.: Visualization and knowledge discovery for high dimensional data. In: *Proceedings of the Second International Workshop on User Interfaces to Data Intensive Systems*. (2001) 5–24
12. Theron, R., Paillard, D., Cortijo, E., Flores, J.A., Vaquero, M., Sierro, F.J., Waelbroeck, C.: Rapid reconstruction of paleoenvironmental features using a new multiplatform program. *Micropaleontology* **50** (2004) 391–395
13. Robinson, A.C., Chen, J., Meyer, E.J., MacEachren, A.M.: Combining usability techniques to design geovisualization tools for epidemiology. *Cartography and Geographic Information Science*, **32** (2005) 243–255

Per-pixel Rendering of Terrain Data

Taek Sang Jeong and JungHyun Han*

Department of Computer Science and Engineering, Korea University, Korea

Abstract. This paper presents a novel approach to terrain rendering, which mostly relies on GPU/shader rather than CPU. The most popular representation for terrain data is uniformly sampled height field. As the height field is stored as a texture map, it is directly accessible by a pixel shader. The pixel shader uses a ray casting algorithm, and the CPU and the vertex shader provide ray information to be passed to the pixel shader. Then, the pixel shader samples the ray, computes the intersection of the ray and the terrain surface, and finally determines the pixel color. The experimental results show the feasibility of the shader-intensive approach to real-time terrain rendering.

1 Introduction

Terrain rendering is essential for many applications such as 3D games, geographic information systems, virtual reality, flight simulation, etc. Applications usually require walk-through or fly-through navigation of the terrain in an *interactive* or *real-time mode*. The most popular representation for terrain data is uniformly sampled *height field*, often called digital elevation model[1]. The height field is a set of height or elevation data sampled in a uniform grid, and its simplicity has made it very popular in terrain visualization. Various approaches have been proposed in terrain rendering, and a recent survey can be found in [2]. In the wide spectrum of the proposed approaches, a common factor is that they all require a tremendous amount of CPU computation, for example, to do LOD(levels of detail) control, occlusion culling, etc.

The role of computer graphics hardware is performing the *rendering pipeline*, which is largely partitioned into *vertex processing* and *fragment processing*. The early graphics hardware handled only the fragment processing, but soon evolved to be able to handle vertex processing which includes transformation and lighting (T&L). Recently, we have witnessed the graphical processing unit (GPU) revolution where the hardware capability has been exponentially growing and further the hardware has become *programmable*. The programmable pipeline is named *shader*. The shader program is a set of GPU instructions, and is partitioned into *vertex shader* (or vertex program) and *pixel shader* (or fragment program).

This paper proposes shader-based terrain rendering algorithms, which maximally utilize the powerful computing capabilities of the shader. Taking the

* Corresponding author.

rendering load off the CPU, the CPU can do otherwise impossible missions, such as AI control in game applications.

2 Why Shader for Terrain Rendering

The original terrain data are often perceived too large to display at interactive frame rates. Therefore, various methods have been adopted that reduce the complexity of the terrain data but retain the image quality. The LOD(levels of detail) method has been popular as a suitable tool that selects only a subset of the height field points and produces a coarser mesh from the subset. For example, the near/important parts are represented by a large number of small polygons, and the far/unimportant parts by a small number of large polygons. Good examples of the LOD-based algorithms are found in [3][4][5].

The graphics hardware revolution makes people call the above-mentioned *CPU-intensive* approaches into question, i.e. the bottleneck in the terrain rendering process often lies in the CPU, which performs terrain simplification and LOD generation *per each frame*. In contrast, the increased computing capability of the GPU allows the GPU to be loaded with more work. Reflecting the changed computing environment, for example, Losasso and Hoppe[6] proposed so-called *geo-clipmap* approach, which adopts a fairly simple LOD scheme such that a finer resolution is generated for a region closer to the viewpoint, and a coarser resolution for a farther region. Then, the simple LOD terrain is passed to GPU, which handles the discontinuities between the different LOD levels of the terrain data.

Along the emerging trend, this paper proposes a shader-intensive terrain rendering approach, where the CPU does little or only a small amount of work, and almost all rendering jobs are assigned to the GPU, more precisely to the pixel shader. The proposed shader-intensive approach presents the terrain data represented as a *texture map* to the pixel shader. Then, the pixel shader renders the terrain data using *ray casting*. The traditional discrete ray casting algorithm is used, where a ray from the viewpoint to a screen pixel position is generated, and then the intersection between the ray and the terrain surface is computed.

3 Rendering Process

The overview of the rendering algorithm can be stated as follows. First of all, *terrain envelope* is computed off-line, which is defined as a coarse superset of the real terrain geometry. The bounding box of the terrain data is a kind of envelope, but is too loose. A *tighter* envelope example is shown in Fig. 1-(a).

In the vertex shader, the *object-space* coordinates of the envelope mesh vertices are output as *texture* values (using TEXCOORD0), and therefore the pixel shader obtains the *interpolated* object-space coordinates for each pixel position. Those coordinates are taken as the starting points for ray casting, as illustrated in Fig. 1-(b). Each ray is then sampled by *uniformly-spaced* points. Each sampled point is processed *in order* towards the terrain surface, as illustrated in

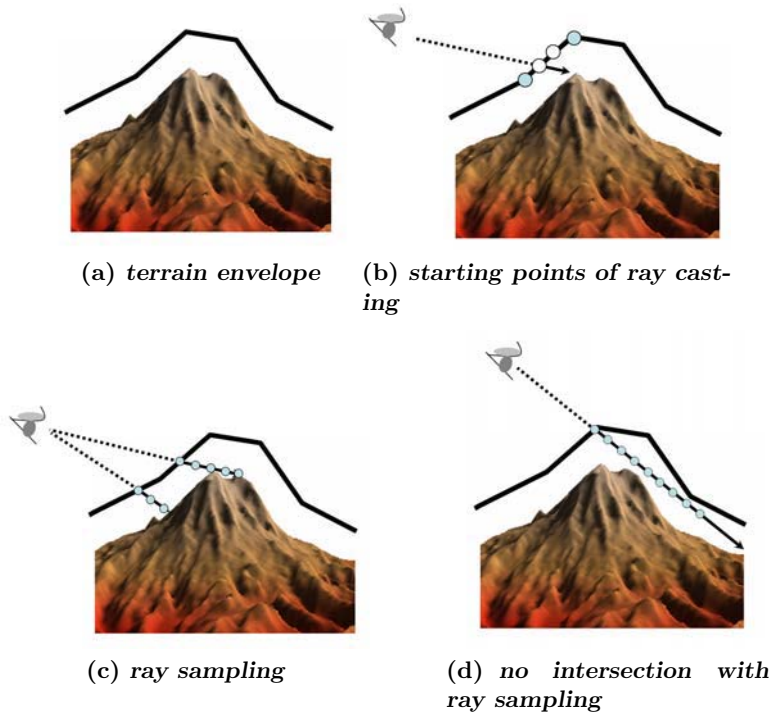


Fig. 1. Terrain envelope and ray casting

Fig. 1-(c). Each sampled point is tested if it lies below the terrain surface. If a point is found to be below the surface, the current and previous points are interpolated to determine the actual intersection position between the ray and the terrain surface.

For the intersection point, lighting is computed, i.e. Phong shading is implemented. For Phong shading, normals at the intersection points are obtained from the *normal map*, which is pre-computed from the terrain height map and passed to the pixel shader. The intersection position's (x,z)-coordinates are used to reference the texture map (real image texture!) of the terrain surface.

4 Rendering Optimization

As demonstrated in Fig. 1-(c), the required numbers of sampled points vary, depending on both the eye position and the ray direction. The major problem of the discrete ray casting is that it is hard to find the upper limit on the number of the sampled points. See Fig. 1-(d), where the ray is sampled for a finite number of iterations, but no intersection is found yet. Do we have to move further? Until when? There is no 'reasonable' upper limit. With a *finite* number of iterations, the intersection point between the ray and the target object is not guaranteed

to be found even when it really exists. Only the *infinite* number of iterations guarantees it, but obviously we cannot do so.

Recently, shader-based ray casting has been explored in the domain of height field, and tackled the sampling problem. For example, Policarpo [7] proposed to use a combination of linear search and binary search, and Donnelly [8] proposed to use a so-called *distance map* in order to non-uniformly sample the ray. However, the intersection point is not guaranteed to be found in the proposed approaches. The rendering algorithm presented in this paper linearly searches the uniformly sampled points, but the intersection point is always found, if any.

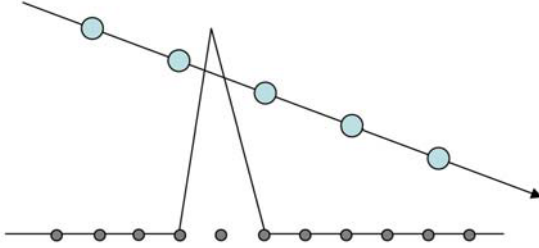


Fig. 2. Ray sampling resolution

Note that the height field data are organized in a *uniform* grid. In the proposed approach, the ray is uniformly sampled exactly at the resolution of the height field data. Then, it is guaranteed that we do not miss an intersection between the ray and the terrain surface. In contrast, it is not guaranteed as long as the ray samples are spaced farther than a single height texel, as shown in Fig. 2.

However, correctly setting the sampling resolution is not enough. We have to find a way to avoid infinite sampling. The solution is found by processing a *bounded* region at a time. For example, the entire height field can be partitioned into 9x9-sized blocks. Then, we need just 9 sampled points per a ray for processing the block. (Recall that the sampling resolution is equal to the height texel resolution.) If an intersection is found during the 9 iterations, color is computed for the intersection point. Otherwise, the ray segment can be safely discarded as the ray will be *re-traced* in the adjacent block(s). In other words, the undetermined pixel position will be processed again when an adjacent block is picked up.

The envelope is pre-computed, and the CPU's role is simply rendering the pre-computed envelope. The *tighter* envelopes we have, the less pixels are processed in the pixel shader. For this purpose, we have adopted the *simplification envelope* approach¹ by Cohen *et al.* [9] Fig. 3-(a) and -(b) show a part of the height field data, and Fig. 3-(c) is its *outer mesh*, which is generated by adaptively offsetting the vertices (all along +y direction) of the original mesh. The original mesh is taken as the *inner mesh*, and then mesh simplification is done only

¹ The original mesh's vertices are offset by a user-specified distance ϵ along its normal and opposite directions to produce *outer* and *inner meshes*, respectively. Then, the original mesh is simplified between the inner and outer meshes. Therefore, the simplified mesh is guaranteed to be within ϵ from the original mesh.

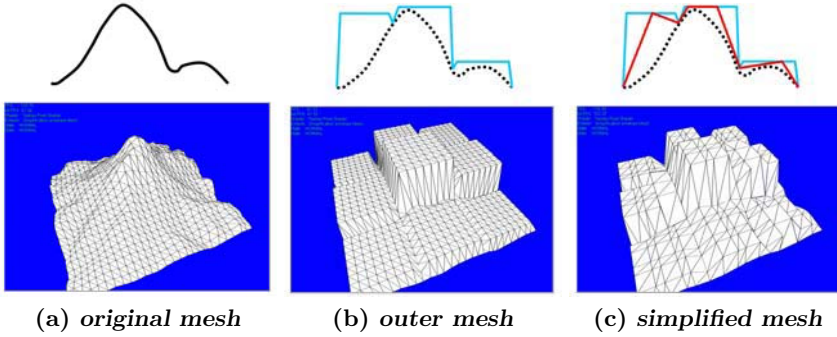


Fig. 3. Envelope construction

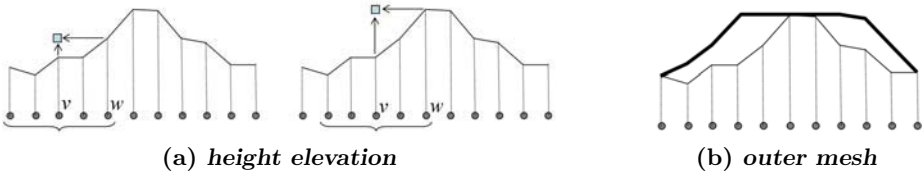


Fig. 4. Outer mesh construction

between the inner and outer meshes. For the sake of simplicity, the outer mesh itself is simplified in the current implementation, and the resulting simplified mesh, shown in Fig. 3-(d), lies between the original mesh (inner mesh) and the ‘fine-resolution’ outer mesh. *Vertex decimation* [10] is currently used as a simplification method.

Let us discuss the outer mesh construction process. See Fig. 4-(a) for an example in the cross section of the height field. Each vertex v of the original height field has to be vertically (along $+y$ direction) offset to generate the outer mesh. To determine v ’s offset distance, consider a square (denoted by a brace in Fig. 4-(a)) centered at v , and take the height field vertex w which is inside the square and has the *highest elevation*. The elevation of v in the outer mesh is set to that of w . Such elevated or vertically offset vertices constitute the outer mesh, as shown in Fig. 4-(b).

Recall that we need *bounded* regions. For the current implementation, the size of a bounded region is 9×9 . The outer mesh construction process is done per a bounded region. Furthermore, note that the *boundary vertices* of each region are not offset, as shown in Fig. 4-(b). By not offsetting the boundary vertices, the *re-traced* ray (discussed above) can be assigned the starting position for discrete ray casting.

5 Experiment Results

The terrain data set used for experiment is 1025×1025 -sized Puget Sound[11]. Fig. 5 shows the height field data and its texture map. The height is represented by 1 byte, and therefore in the range of $[0,255]$. Fig. 5-(a) is the gray-scale

visualization of the height field data, where black(0) represents the lowest elevation and white(255) represents the highest elevation.

The experiment is implemented on Pentium 4 (3.0 GHz) and NVidia GeForce 6800 Ultra with DirectX 9.0c and pixel shade ps_3_0. Unlike the previous profiles, ps_3_0 supports *static* and *dynamic flow control* such as iteration and branch, and therefore sampling of the ray becomes possible. Fig. 6 shows the snapshots of the rendered terrain from different viewpoints. The average frame rate is 78 fps (frames per second).

Fig. 7 compares two images: one is rendering with the polygon mesh obtained from the height field data, and the other is rendering with ray casting. Both of them have the comparable image quality.

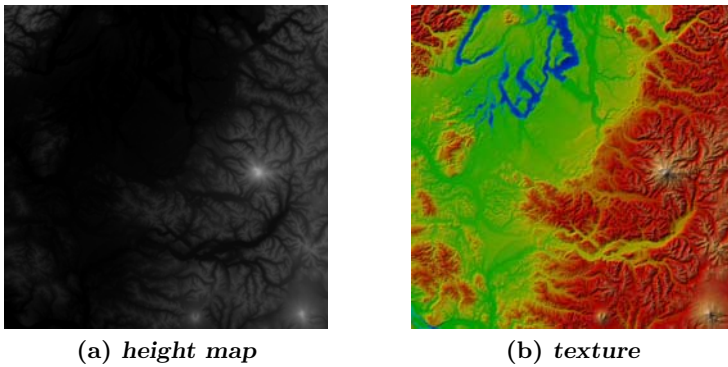


Fig. 5. Test data

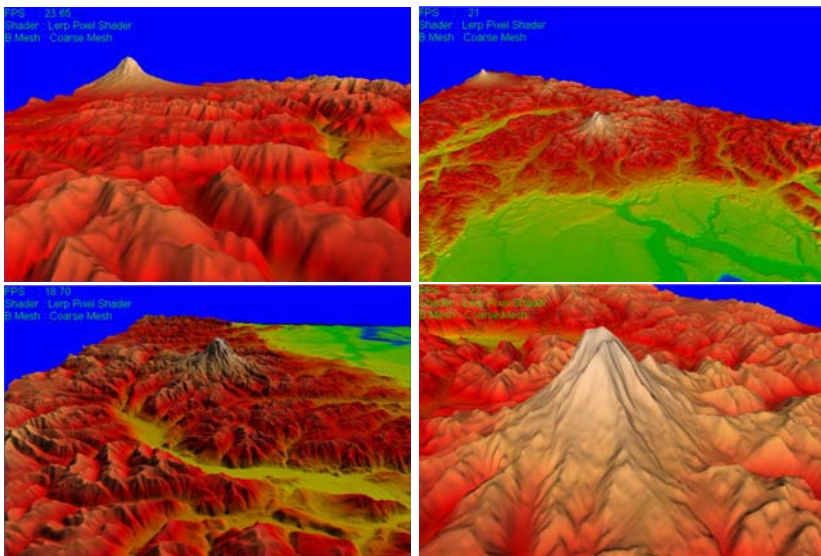
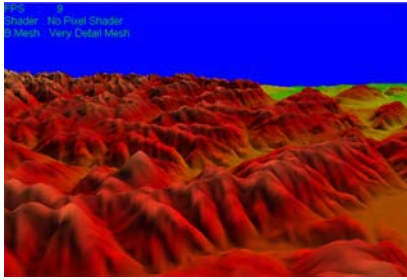
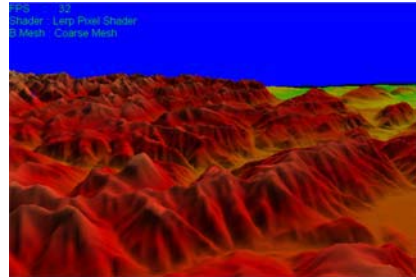


Fig. 6. Rendered images

(a) *mesh rendering*(b) *ray casting***Fig. 7.** Rendering result comparison

6 Conclusion

This paper presents a novel shader-intensive approach to terrain rendering. The height field stored as a texture map is directly accessible by a pixel shader. The pixel shader uses a ray casting algorithm, and the CPU and the vertex shader provide ray information to be passed to the pixel shader. The experimental results show the feasibility of the shader-intensive approach to real-time terrain rendering.

The earnest work of shader-based terrain rendering has been reported in [12] and [13]. The work by [12] proposes a simple application of shader. In the work by [13], however, the rendering process mostly relies on GPU/shader rather than CPU. Our approach follows such a *shader-intensive* approach as [13].

In the current implementation, the CPU does little work for the shader. However, the authors envision that the CPU could perform terrain data simplification and LOD generation unless they cause a bottleneck in the rendering process. Then, the simplified portion could be accommodated by a *mipmap* structure. Such a small CPU program can be plugged in the proposed framework² where the main body of rendering is performed by the shader.

Acknowledgements

This research was supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005-(C1090-0501-0019). This work was also sponsored and funded by Korea Game Development & Promotion Institute as Korean government project (Ministry of Culture and Tourism).

² The traditional LOD construction algorithms cannot be moved to shader because the neighbouring vertex information which is essential for an LOD construction is not available to shaders.

References

1. U.S. Geological Survey. <http://www.usgs.gov/>
2. R. Pajarola: Overview of Quadtree-based Terrain Triangulation and Visualization. UCI-ICS Technical Report No. 02-01, Department of Information & Computer Science, University of California, Irvine, 2002.
3. P. Lindstrom, V. Pascucci: Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization. IEEE Transactions on Visualization and Computer Graphics, Vol. 8, No. 3, July-September, pp. 239-254, 2002.
4. M. Duchaineau, M. Wolinsky, D. Sigeti, M. Miller, C. Aldrich, M. Mineev-Weinstein: ROAMing Terrain: Real-time Optimally Adapting Meshes. IEEE Visualization 1997, pp. 81-88, 1997.
5. S. Röttger, W. Heidrich, P. Slusallek, H.-P. Seidel: Real-time Generation of Continuous Levels of Detail for Height Fields. Central Europe Conf. on Computer Graphics and Vis., pp. 315-322, 1998.
6. F. Losasso, H. Hoppe: Geometry Clipmaps: Terrain Rendering using Nested Regular Grids. ACM SIGGRAPH, pp. 769-776, 2004.
7. F. Policarpo: Relief Mapping in a Pixel Shader using Binary Search. <http://www.paralelo.com.br/arquivos/ReliefMapping.pdf>.
8. W. Donnelly: Per-pixel Displacement Mapping with Distance Functions. GPU Gems 2, Addison-Wesley, pp. 123-136, 2005.
9. J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber: Simplification Envelopes. ACM SIGGRAPH, pp. 119-128, 1996.
10. W. Schroeder, J. Zarge, W. Lorensen: Decimation of Triangle Meshes. ACM SIGGRAPH, pp. 65-70, 1992.
11. Puget Sound. http://www.cc.gatech.edu/projects/large_models/ps.html
12. Aaron Burton: Voxel Rendering with PS_3_0. ShaderX2 Shader Programming Tips & Tricks with DirectX9, Wordware publishing, pp.161-171, 2004.
13. A. Asirvatham, H. Hoppe: Terrain Rendering using GPU-based Geometry Clipmaps. GPU Gems 2, Addison-Wesley, pp. 27-45, 2005.

Spherical Harmonic Transforms Using Quadratures and Least Squares

J.A.R. Blais¹ and M.A. Soofi²

^{1,2}Department of Geomatics Engineering

¹Pacific Institute for the Mathematical Sciences

University of Calgary, Calgary, AB, T2N 1N4, Canada

blais@ucalgary.ca, soofi@ucalgary.ca

www.ucalgary.ca/~blais

Abstract. Spherical Harmonic Transforms (SHTs) which are essentially Fourier transforms on the sphere are critical in global geopotential and related applications. For analysis purposes, discrete SHTs are difficult to formulate for an optimal discretization of the sphere, especially for applications with requirements in terms of near-isometric grids and special considerations in the polar regions. With the enormous global datasets becoming available from satellite systems, very high degrees and orders are required and the implied computational efforts are very challenging. Among the best known strategies for discrete SHTs are quadratures and least squares. The computational aspects of SHTs and their inverses using both quadrature and least-squares estimation methods are discussed with special emphasis on information conservation and numerical stability. Parallel and grid computations are imperative for a number of geodetic, geophysical and related applications, and these are currently under investigation.

1 Introduction

Domains with spherical topology are very common in astronomy, cosmology, geophysics, geodesy and related disciplines. On the spherical Earth as on the celestial sphere, computations can be done for regional and global domains using planar and spherical formulations. Spherical quadratures and least-squares estimation are used to convert continuous integral formulations into summations over data lattices. Spherical topologies are quite different from planar ones and these have important implications in the computational aspects of data processing.

Spherical geocomputations for regional domains of even continental extents can be reduced to planar computations and under assumptions of stationarity or shift invariance, discrete regular array computations can be optimized using Fast Fourier Transforms (FFTs). Specifically, convolution operations for filtering and other data processing applications thereby require only $O(N\log N)$ instead of $O(N^2)$ operations for N data in one dimension, $O(N^2\log N)$ instead of $O(N^4)$ operations for $N \times N$ data in two dimensions, and so on.

For global applications, Gaussian, equiangular and other similar rectangular grids can be used for spherical quadratures and discrete convolutions. Various quadrature

strategies are available in the literature going back to Gauss and Neumann, in addition to least-squares estimation techniques (e.g. [14]). Other approaches have also been used for discretization and analysis of functions on the sphere using icosahedral, triangular and curvilinear tessellations. Depending on the applications, these strategies may be preferable to the rectangular ones which will be discussed in the following.

2 Continuous and Discrete SHTs

The orthogonal or Fourier expansion of a function $f(\theta, \lambda)$ on the sphere \mathbf{S}^2 is given by

$$f(\theta, \lambda) = \sum_{n=0}^{\infty} \sum_{|m| \leq n} f_{n,m} Y_n^m(\theta, \lambda) \quad (1)$$

using colatitude θ and longitude λ , where the basis functions $Y_n^m(\theta, \lambda)$ are called the spherical harmonics satisfying the (spherical) Laplace equation $\Delta_{\mathbf{S}^2} Y_n^m(\theta, \lambda) = 0$, for all $|m| \leq n$ and $n = 0, 1, 2, \dots$. This is an orthogonal decomposition in the Hilbert space $L^2(\mathbf{S}^2)$ of functions square integrable with respect to the standard rotation invariant measure $d\sigma = \sin \theta d\theta d\lambda$ on \mathbf{S}^2 . In particular, the Fourier or spherical harmonic coefficients appearing in the preceding expansion are obtained as inner products

$$f_{n,m} = \int_{\mathbf{S}^2} f(\theta, \lambda) \bar{Y}_n^m(\theta, \lambda) d\sigma \quad (2)$$

with the overbar denoting the complex conjugate. In most practical applications, the functions $f(\theta, \lambda)$ are band-limited in the sense that only a finite number of those coefficients are nonzero, i.e. $f_{n,m} \equiv 0$ for all $n \geq N$ and orders $|m| \leq n$. Hence, using the regular grid $\theta_j = j\pi/2N$ and $\lambda_k = k\pi/N$, $j, k = 0, \dots, 2N-1$, spherical harmonic synthesis can be formulated as

$$f(\theta_j, \lambda_k) = \sum_{n=0}^{N-1} \sum_{|m| \leq n} f_{n,m} Y_n^m(\theta_j, \lambda_k) \quad (3)$$

and using some appropriate spherical quadrature, the corresponding spherical harmonic analysis can be formulated as

$$f_{n,m} = \sum_{j=0}^{2N-1} \sum_{k=0}^{2N-1} q_j f(\theta_j, \lambda_k) \bar{Y}_n^m(\theta_j, \lambda_k) \quad (4)$$

for quadrature weights q_j as discussed by various authors e.g. [2], [3], [4] and [6].

The usual geodetic spherical harmonic formulation is given as

$$f(\theta, \lambda) = \sum_{n=0}^{\infty} \sum_{m=0}^n [\ddot{C}_{nm} \cos m\lambda + \ddot{S}_{nm} \sin m\lambda] \ddot{P}_{nm}(\cos \theta) \quad (5)$$

where

$$\left\{ \begin{array}{c} \ddot{C}_{nm} \\ \ddot{S}_{nm} \end{array} \right\} = \frac{1}{4\pi} \int_{\mathbf{S}^2} f(\theta, \lambda) \left\{ \begin{array}{c} \cos m\lambda \\ \sin m\lambda \end{array} \right\} \ddot{P}_{nm}(\cos \theta) d\sigma \quad (6)$$

and

$$\begin{aligned}\ddot{P}_{nm}(\cos \theta) &= \sqrt{\frac{2(2n+1)(n-m)!}{(n+m)!}} P_{nm}(\cos \theta) \\ \ddot{P}_n(\cos \theta) &= \sqrt{2n+1} P_n(\cos \theta)\end{aligned}\quad (7)$$

are the normalized associated Legendre functions. These associated Legendre functions are related to the preceding spherical harmonic functions $Y_n^m(\theta, \lambda)$ as follows

$$Y_n^m(\theta, \lambda) = (-1)^m \sqrt{\frac{(2n+1)(n-m)!}{4\pi(n+m)!}} P_{nm}(\cos \theta) e^{-im\lambda} \quad (8)$$

and geodetic normalization is slightly different from the usual mathematical normalization (see e.g. [5] and [8]).

Explicitly, using the geodetic formulation and convention, one has for synthesis, given normalized spherical harmonic coefficients a_{nm} and b_{nm} for $m \leq n$, $n = 0, 1, 2, \dots, N-1$,

$$\begin{aligned}f(\theta, \lambda) &= \sum_{m=0}^{N-1} \sum_{n=m}^{N-1} (a_{nm} \cos m\lambda + b_{nm} \sin m\lambda) \ddot{P}_{nm}(\cos \theta) \\ &= \sum_{m=0}^{N-1} \left\{ \left(\sum_{n=m}^{N-1} a_{nm} \ddot{P}_{nm}(\cos \theta) \right) \cos m\lambda + \left(\sum_{n=m}^{N-1} b_{nm} \ddot{P}_{nm}(\cos \theta) \right) \sin m\lambda \right\}\end{aligned}\quad (9)$$

and defining

$$\begin{Bmatrix} A_m(\theta) \\ B_m(\theta) \end{Bmatrix} = \sum_{n=m}^{N-1} \begin{Bmatrix} a_{nm} \\ b_{nm} \end{Bmatrix} \ddot{P}_{nm}(\cos \theta) \quad (10)$$

one has

$$\begin{aligned}f(\theta, \lambda) &= \sum_{m=0}^{N-1} \{A_m(\theta) \cos m\lambda + B_m(\theta) \sin m\lambda\} \\ &= \text{Re IDFT} [A_m(\theta) + i B_m(\theta)]\end{aligned}\quad (11)$$

assuming discrete longitudes $\lambda_k = k\pi/N$, $k=0, 1, 2, \dots, 2N-1$, for unspecified discrete colatitudes θ . Writing $C_m(\theta) = A_m(\theta) + iB_m(\theta)$, and correspondingly $c_{nm} = a_{nm} + ib_{nm}$, one then has

$$f(\theta, \lambda_k) = \text{Re IDFT}_k [C_m(\theta)] \quad (12)$$

where

$$C_m(\theta) = \sum_{n=m}^{N-1} (a_{nm} + ib_{nm}) \ddot{P}_{nm}(\cos \theta) = \sum_{n=m}^{N-1} c_{nm} \ddot{P}_{nm}(\cos \theta) \quad (13)$$

in which IDFT stands for inverse discrete Fourier transform. For analysis, given data $f(\theta_j, \lambda_k)$ at $\theta_j = j\pi/2N$ and $\lambda_k = k\pi/N$, $j, k = 0, 1, 2, \dots, 2N-1$, the normalized spherical harmonic coefficients a_{nm} and b_{nm} for $m \leq n$, $n = 0, 1, 2, \dots, N-1$, can be evaluated as follows:

$$\begin{Bmatrix} a_{nm} \\ b_{nm} \end{Bmatrix} = \frac{\pi(-1)^m}{N} \sum_{j=0}^{2N-1} \sum_{k=0}^{2N-1} q_{jk} f(\theta_j, \lambda_k) \begin{Bmatrix} \cos m\lambda_k \\ \sin m\lambda_k \end{Bmatrix} \ddot{P}_{nm}(\cos \theta_j) \quad (14)$$

or, using complex coefficients,

$$\begin{aligned} c_{nm} &= \frac{\pi(-1)^m}{N} \sum_{j=0}^{2N-1} \sum_{k=0}^{2N-1} q_j f(\theta_j, \lambda_k) e^{+im\lambda_k} \ddot{P}_{nm}(\cos \theta_j) \\ &= \pi \sum_{j=0}^{2N-1} (-1)^m q_j \ddot{P}_{nm}(\cos \theta_j) \text{DFT}_k[f(\theta_j, \lambda_k)] \end{aligned} \quad (15)$$

with the preceding quadrature weights q_j . Notice that in practice, depending on conventions, DFT and IDFT could be interchanged in the preceding derivation and for computational efficiency, direct and inverse FFTs would be substituted.

The preceding derivation can easily be modified for data grids with $\Delta\theta = \Delta\lambda$ such as

$$\{(\theta_j, \lambda_k) \mid \theta_j = j\pi/2N, \lambda_k = k\pi/2N; j = 0, 1, \dots, 2N-1, k = 0, 1, \dots, 4N-1\}$$

with the possible exclusion of the poles as

$$\{(\theta_j, \lambda_k) \mid \theta_j = (j+1/2)\pi/2N, \lambda_k = k\pi/2N; j = 0, 1, \dots, 2N-1, k = 0, 1, \dots, 4N-1\}$$

which also permit the use of hemispherical symmetries in the associated Legendre functions

$$P_{nm}(\cos(\pi - \theta)) = (-1)^{n+m} P_{nm}(\cos \theta). \quad (16)$$

More details can be found in [4]. Notice that these data grids have $2N \times 4N$ or $8N^2$ quantities for N^2 spectral coefficients for band limitedness N .

3 Spectral Analysis Using Least Squares

Let us consider the regular data grid

$$\{(\theta_j, \lambda_k) \mid \theta_j = j\pi/N, \lambda_k = k\pi/2N; j = 0, 1, \dots, N-1, k = 0, 1, \dots, 2N-1\}$$

or with the exclusion of the poles,

$$\{(\theta_j, \lambda_k) \mid \theta_j = (j+1/2)\pi/N, \lambda_k = k\pi/2N; j = 0, 1, \dots, N-1, k = 0, 1, \dots, 2N-1\}$$

which implies $\Delta\theta = \Delta\lambda$. This is the grid normally considered in geodesy to model the geopotential field with least data for a spectral expansion to degree and order $N - 1$.

From the preceding discussion, one can write

$$f(\theta_j, \lambda_k) = \text{Re IDFT}_k \left[\sum_{n=m}^{N-1} (a_{nm} + ib_{nm}) \ddot{P}_{nm}(\cos \theta_j) \right] \quad (17)$$

for the unknown a_{nm} and b_{nm} . Hence

$$\text{DFT}_k[f(\theta_j, \lambda_k) + i0] = \sum_{n=m}^{N-1} (a_{nm} + ib_{nm}) \ddot{P}_{nm}(\cos \theta_j) \quad (18)$$

which implies that for each $m = 0, 1, \dots, N-1$, least squares can be used to estimate the spectral coefficients $a_{mm}, a_{m+1,m}, \dots, a_{N-1,m}$ and $b_{mm}, b_{m+1,m}, \dots, b_{N-1,m}$.

It is very important to notice that a maximum degree of $N - 1$ is critical for the least-squares solution for the spectral coefficients $a_{nm}, a_{m+1,m}, \dots, a_{N-1,m}$ and $b_{nm}, b_{m+1,m}, \dots, b_{N-1,m}$ using N partitions in latitude, i.e. $\theta_j = j\pi/N$, or $\theta_j = (j+1/2)\pi/N$, $j =$

0, 1, ..., N- 1. In other words, the use of N parallels for degree N would lead to an underdetermined system for a_{nm} , $a_{m+1,m}$, ..., $a_{N,m}$ and b_{nm} , $b_{m+1,m}$, ..., $b_{N,m}$, as apparently suggested in [14].

In most practical implementations, the partition in latitude is shifted by half a grid interval to avoid any numerical complication at the pole for $\theta = 0$. Furthermore, the partition in latitude is not required to be equispaced for the least-squares estimation of the spectral coefficients. However, in longitude, equispacing is required for the usual implementation of the DFTs as FFTs.

4 Numerical Experimentation

As indicated earlier, there are several formulations for employing spherical harmonics as an analysis tool. One popular code that is readily available is Spherpac [1] of which a new version has been released recently. Other experimental codes are those of Driscoll and Healy [6] and the follow-ons, such as [7] and [12], plus the example algorithm described by Mohlenkamp [9][10] and offered as a partial sample implementation in [11]. Experimentation with these codes has shown scaling differences from that which is expected in a geodetic context [13].

Using the Driscoll and Healy [6] formulation modified as described in Section 2, extensive experimentation using different grids on several computer platforms in double precision (i.e. REAL*8) and quadruple precision (i.e. REAL*16) has been carried out. However, only double precision results are reported in this paper. The first synthesis started with unit coefficients, $a_{nm} = b_{nm} = 1$, except for $b_{n0} = 0$, for all degrees n and orders m, which corresponds to white noise. Then, following analysis of the generated spatial grid values, the coefficients are recomputed and root-mean-square (RMS) values are given for this synthesis/analysis. Then after another synthesis using these recomputed coefficients, RMS values of recomputed grid residuals are given for the second synthesis. Hence, starting with arbitrary coefficients $\{c_{nm}\}$, the procedure can be summarized as follows:

$SHT[SHT^{-1}\{\{c_{nm}\}\}] - \{\{c_{nm}\}\} \rightarrow$ RMS of first synthesis/analysis,

and

$SHT^{-1}[SHT[SHT^{-1}\{\{c_{nm}\}\}]] - SHT^{-1}\{\{c_{nm}\}\} \rightarrow$ RMS of second synthesis.

Notice that the first RMS is in the spectral domain while the second is in the spatial domain. The SHTs and SHT⁻¹s are evaluated and re-evaluated explicitly to study their numerical stability and the computational efficiency. Table 1(top) lists the RMS values and the computation times with grids $\Delta\theta = \frac{1}{2}\Delta\lambda$ (i.e., 2Nx2N) and $\Delta\theta = \Delta\lambda$ (i.e., Nx2N and 2Nx4N). The above procedure is repeated for coefficients corresponding to $1/\text{degree}^2$, i.e. explicitly, $a_{nm} = 1/(n+1)^2$, $b_{nm} = 0$ for $m=0$, and $1/(n+1)^2$, otherwise, for all degrees n and orders m, which simulate a physically realizable situation. The corresponding RMS values and computation times for various grids are listed in the bottom part of Table 1. It should be noted that the results for the grid Nx2N are based on least squares whereas for 2Nx2N and 2Nx4N the quadrature scheme discussed in Section 2 is used, and the computations were done on an AMD 64 Athlon FX-51 PC in REAL*8 precision.

Table 1. SHT RMS values and computation times for Nx2N, 2Nx2N and 2Nx4N grids, where N-1 is the maximum degree of expansion. Results for the Nx2N grid are obtained using the least-squares technique and for 2Nx2N and 2Nx4N, the quadrature scheme. **Top:** Input: unit spectral coefficients. **Bottom:** Input: 1/degree² spectral coefficients.

	Degrees	Grid	Synthesis/Analysis		Synthesis	
			RMS(coef.)	Time(sec.)	RMS(data)	Time(sec.)
N x 2N	0-63	64x128	2.632e-15	0.03	1.123e-13	0.02
	0-127	128x256	5.354e-15	0.44	4.368e-13	0.06
	0-255	256x512	1.062e-14	6.66	1.743e-12	0.53
	0-511	512x1024	2.126e-14	226.06	7.045e-12	4.20
	0-1023	1024x2048	4.245e-14	10290.73	2.796e-11	32.94
	0-1499	1500x3000	6.046e-14	15552.55	5.808e-11	102.23
2N x 2N	0-63	128x128	5.966e-15	0.01	7.763e-15	0.01
	0-127	256x256	1.693e-14	0.08	4.315e-14	0.07
	0-255	512x512	5.000e-14	0.69	1.906e-13	0.59
	0-511	1024x1024	6.879e-14	6.19	2.943e-13	5.12
	0-1023	2048x2048	2.817e-13	46.30	2.819e-12	37.99
	0-1499	3000x3000	7.121e-13	142.29	9.497e-12	117.34
2N x 4N	0-63	128x256	5.978e-15	0.01	7.844e-15	0.01
	0-127	256x512	1.686e-14	0.08	3.605e-14	0.07
	0-255	512x1024	4.994e-14	0.74	1.906e-13	0.63
	0-511	1024x2048	6.874e-14	6.32	2.787e-13	5.22
	0-1023	2048x4096	2.817e-13	47.54	2.818e-12	38.60
	0-1499	3000x6000	7.121e-13	146.22	9.497e-12	120.03
N x 2N	0-63	64x128	1.718e-17	0.05	7.765e-16	0.00
	0-127	128x256	2.253e-17	0.47	1.797e-15	0.08
	0-255	256x512	1.675e-17	6.95	2.804e-15	0.55
	0-511	512x1024	1.078e-17	247.97	3.583e-15	4.66
	0-1023	1024x2048	7.151e-18	11051.94	4.780e-15	35.67
	0-1499	1500x3000	6.177e-18	16728.81	6.167e-15	112.25
2N x 2N	0-63	128x128	4.142e-17	0.01	3.844e-17	0.01
	0-127	256x256	5.806e-17	0.08	9.084e-17	0.07
	0-255	512x512	5.201e-17	0.74	1.212e-16	0.59
	0-511	1024x1024	2.877e-17	6.45	5.578e-17	5.12
	0-1023	2048x2048	3.437e-17	48.90	2.097e-16	38.02
	0-1499	3000x3000	5.057e-17	151.04	5.208e-16	117.30
2N x 4N	0-63	128x256	4.120e-17	0.01	3.413e-17	0.01
	0-127	256x512	5.774e-17	0.09	6.358e-17	0.07
	0-255	512x1024	5.175e-17	0.79	1.185e-16	0.64
	0-511	1024x2048	2.876e-17	6.66	7.386e-17	5.19
	0-1023	2048x4096	3.438e-17	50.32	2.041e-16	38.53
	0-1499	3000x6000	5.057e-17	155.86	5.058e-16	119.97

The RMS of errors involved in the least squares ($N \times 2N$) and quadratures ($2N \times 2N$ and $2N \times 4N$) schemes are comparable (Table 1) and varying in the range 10^{-12} to 10^{-15} for unit coefficients and 10^{-15} to 10^{-17} for $1/\text{degree}^2$ coefficients. The computation times for both the approaches, however, vary significantly. Preliminary investigations indicate inefficiency in the least-squares approach resulting in extra efforts needed in the normal equations corresponding to each order m to estimate the related spectral coefficients. Work is in progress to optimize this part of the code to make the computation times comparable to the quadrature scheme.

Spectral analysis of the synthesis/analysis results can be done degree by degree to study the characteristics of the estimated spectral harmonic coefficients. The results of the second synthesis also enable a study of the spatial results parallel by parallel, especially for the polar regions. Such investigations are currently underway to better characterize the numerical stability and reliability of the SHT and SHT⁻¹. Ongoing experimentation is attempting to carry out the computations to higher degrees and orders.

5 Concluding Remarks

Considerable work has been done on solving the computational complexities, and enhancing the speed of calculation of spherical harmonics transforms for different equiangular grids. The approach of Driscoll and Healy [6] is exact for exact arithmetic, and with a number of modifications, different implementations have been experimented with, leading to RMS errors of orders 10^{-12} to 10^{-15} with unit coefficients of degrees and orders up to 1500. Comparable RMS have been obtained with the least-squares estimation approach. However the computational efforts with least squares are quite significantly larger than with the quadrature approach. When starting with spherical harmonic coefficients corresponding to $1/\text{degree}^2$, the previously mentioned analysis and synthesis results are improved to 10^{-15} - 10^{-18} with both approaches. The latter simulations are perhaps more indicative of the expected numerical accuracies in practice.

All these computational experiments started with unit and $1/\text{degree}^2$ coefficients of degrees 0, 1, 2, ..., $N-1$ for spherical $N \times 2N$ grids with least squares, and $2N \times 2N$ and $2N \times 4N$ grids with quadratures, offset in latitude by half a grid unit from the pole. It remains to be seen how these spatial lattices differ for the same spectrum of degree and order $N - 1$ for a specified N . This comparative spatial analysis is currently underway.

Computations for higher degrees and orders are under consideration assuming the availability of parallel FFT code in quadruple precision. As enormous quantities of data are involved the intended gravity field applications, parallel and grid computations are imperative for these applications. Preliminary experimentation with parallel processing has already been done [15].

Acknowledgements

The authors would like to acknowledge the sponsorship of the Natural Science and Engineering Research Council in the form of a Research Grant to the first author on Computational Tools for the Geosciences. Special thanks are hereby expressed to

Dr. D. Phillips of Information Technologies, University of Calgary, for helping with the optimization of our code for different computer platforms. Comments and suggestions from a colleague, Dr. N. Sneeuw, are also gratefully acknowledged.

References

1. Adams, J.C. and P.N. Swarztrauber [1997]: SPHEREPACK 2.0: A Model Development Facility. <http://www.scd.ucar.edu/softlib/SPHERE.html>
2. Blais, J.A.R. and D.A. Provens [2002]: Spherical Harmonic Analysis and Synthesis for Global Multiresolution Applications. *Journal of Geodesy*, vol.76, no.1, pp.29-35.
3. Blais, J.A.R. and D.A. Provens [2003]: Optimization of Computations in Global Geopotential Field Applications. *Computational Science – ICCS 2003, Part II*, edited by P.M.A. Sloot, D. Abramson, A.V. Bogdanov, J.J. Dongarra, A.Y. Zomaya and Y.E. Gorbachev. *Lecture Notes in Computer Science*, vol.2658, pp.610-618. Springer-Verlag.
4. Blais, J.A.R., D.A. Provens and M.A. Soofi [2005]: Optimization of Spherical Harmonic Transform Computations, in *ICCS 2005*, V.S. Sunderam et al. (eds.), *Lecture Notes in Computer Science (LNCS)*, vol. 3514, p. 74-81.
5. Colombo, O. [1981]: Numerical Methods for Harmonic Analysis on the Sphere. Report no. 310, Department of Geodetic Science and Surveying, The Ohio State University
6. Driscoll, J.R. and D.M. Healy, Jr. [1994]: Computing Fourier Transforms and Convolutions on the 2-Sphere. *Advances in Applied Mathematics*, 15, pp. 202-250.
7. Healy, D., Jr., D. Rockmore, P. Kostelec and S. Moore [1998]: FFTs for the 2-Sphere - Improvements and Variations, To appear in *Advances in Applied Mathematics*, Preprint from <http://www.cs.dartmouth.edu/~geelong/publications> (June 1998).
8. Heiskanen, W.A. and H. Moritz [1967]: *Physical Geodesy*, W.H. Freeman and Company, San Francisco, 363pp.
9. Mohlenkamp, M.J. [1997]: A Fast Transform for Spherical Harmonics. PhD thesis, Yale University.
10. Mohlenkamp, M.J. [1999]: A Fast Transform for Spherical Harmonics. *The Journal of Fourier Analysis and Applications*, 5, 2/3, pp. 159-184, Preprint from <http://amath.colorado.edu/faculty/mjm>.
11. Mohlenkamp, M.J. [2000]: Fast spherical harmonic analysis: sample code. <http://amath.colorado.edu/faculty/mjm>.
12. Moore, S., D. Healy, Jr., D. Rockmore and P. Kostelec [1998]: SpharmonKit25: Spherical Harmonic Transform Kit 2.5, <http://www.cs.dartmouth.edu/~geelong/sphere/>.
13. Provens, D.A. [2003]: Earth Synthesis: Determining Earth's Structure from Geopotential Fields, Unpublished PhD thesis, University of Calgary, Calgary.
14. Sneeuw, N. [1994]: Global Spherical Harmonic Analysis by Least-Squares and Numerical Quadrature Methods in Historical Perspective. *Geophys. J. Int.* 118, 707-716.
15. Soofi, M.A. and J.A.R. Blais [2005]: Parallel Computations of Spherical Harmonic Transforms, Oral presentation at the Annual Meeting of the Canadian Geophysical Union, Banff, Alberta, Canada.

Numerical Simulations of Space-Time Conditional Random Fields of Ground Motions

Robert Jankowski

Faculty of Civil and Environmental Engineering, Gdańsk University of Technology,
ul. Narutowicza 11/12, 80-952 Gdańsk, Poland
jankowr@pg.gda.pl

Abstract. The aim of the present paper is to propose a method of conditional stochastic simulation of propagation of seismic wave using the spatiotemporal correlation function. The method has been used to generate unknown time histories at various points of ground motion random field based on the specified earthquake record at one location. The results of the study show that the method considered gives relatively low simulation errors.

1 Introduction

Simulation of spatiotemporal variation of earthquake field by the deterministic approach is very complex. It requires detailed knowledge of the fault size, rupture mechanisms, propagation paths, distance from the epicentre, local geological and topographical conditions. Therefore in earthquake engineering, the stochastic approach has been proved to be very useful and successfully applied in practice [4].

The incorporation of the seismic wave propagation effect is especially important when large structures such as long bridges, dams, life-line systems and large buildings, are analyzed (see [3], for example). This is due to the fact that their dynamic response can be significantly altered due to the spatial variations of ground motion causing different seismic input acting on different supports of the structure. Detailed analysis of the response of large structures often requires the generation of acceleration records for different locations of supports based on the specified record. This can be achieved by applying the conditional stochastic modelling approach (see [1], [2]).

One of the problems of methods for random field simulation of ground motions concerns the issue of identifying the time correlation of the field. This problem is often simplified by the use of time-delay parameters, which are applied to shift earthquake records in time for different locations (see [2], [3]). It requires the specification of direction of the seismic wave propagation, what is reasonable in the case of line structures but may be problematic in other cases. Moreover, the simulation results obtained in this way might depend on a value of the time step used in the analysis [2].

The aim of the present paper is to propose a method of conditional stochastic simulation of ground motions using the spatiotemporal correlation function. The method has been used to generate unknown acceleration time histories at various field points based on the specified earthquake record at one location.

2 Conditional Random Field Simulation Method

Let us define a correlation in space and time of the ground motion field by a spatio-temporal correlation function of the form (compare with [2]),

$$K(\mathbf{r}_{ij}, t_{ij}) = \sigma^2 \exp\left(-\frac{\omega_d |\mathbf{r}_{ij}|}{2\pi\nu d}\right) \exp(-\beta \Delta t_{ij}), \quad (1)$$

where σ is a standard deviation of the earthquake record, ω_d is a predominant frequency of ground motion, $|\mathbf{r}_{ij}|$ is a distance between the field points i, j , ν is a mean apparent seismic wave velocity in the field, d is a space scale parameter ($d > 0$) depending on local ground conditions, Δt_{ij} is a time lag between the values in the ground motion records for the field points i, j and β is a time scale parameter ($\beta > 0$) describing the degree of time correlation of the field. The scale parameters d and β should be obtained experimentally based on the data concerning the seismic wave propagation patterns in the ground. It should be mentioned, that defining the correlation of the ground motion field by the use of the spatiotemporal correlation function of Eq. (1) does not involve the necessity of specifying the direction of the seismic wave propagation and of selecting the appropriate value of the time step in the earthquake record.

In the random fields theory, the earthquake acceleration record is usually treated as a set of measured values, which satisfy zero-mean Gaussian distribution. A formula for the probability density of Gaussian conditional distribution can be written as [1],

$$f(\mathbf{x}|\mathbf{y}) = (\det \mathbf{K}_c)^{-\frac{1}{2}} \cdot (2\pi)^{-\frac{n}{2}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m}_c)^T \mathbf{K}_c^{-1} (\mathbf{x} - \mathbf{m}_c)\right), \quad (2)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is a n -dimensional vector of unknown values, $\mathbf{y} = [y_1, y_2, \dots, y_{n'}]^T$ is a n' -dimensional vector of known values, \mathbf{K}_c is a conditional covariance matrix,

$$\mathbf{K}_c = \mathbf{K}_{11} - \mathbf{K}_{12} \mathbf{K}_{22}^{-1} \mathbf{K}_{21}, \quad (3)$$

and \mathbf{m}_c is a vector of conditional mean values,

$$\mathbf{m}_c = \mathbf{K}_{12} \mathbf{K}_{22}^{-1} \mathbf{y}, \quad (4)$$

where \mathbf{K}_{11} , \mathbf{K}_{12} , \mathbf{K}_{21} , \mathbf{K}_{22} , are the elements of the covariance matrix of the field determined for all (known and unknown) values,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}_{(n+n') \times (n+n')}. \quad (5)$$

The following numerical procedure, based on the acceptance-rejection theorem (compare with [1], [2]), has been used in this paper for the simulation purposes:

1. Determination of the covariance matrix of the field, \mathbf{K} , according to Eq.(5) assuming the spatiotemporal correlation function defined by Eq.(1).
2. Calculation of the conditional covariance matrix, \mathbf{K}_c , from Eq.(3).

Points 3-6 are repeated for all time steps in the earthquake record:

3. Generation of an unknown vector \mathbf{x} , $x_i = -a_{\max} + 2a_{\max} \cdot r_d$, where a_{\max} is a peak ground acceleration and r_d is a random variable from interval $\langle 0,1 \rangle$.
4. Calculation of vector of conditional mean values, \mathbf{m}_c , according to Eq.(4), and value of conditional density function, $f(\mathbf{x}|\mathbf{y})$, defined by Eq.(2).
5. Generation of a random value, R ,

$$R = (\det \mathbf{K}_c)^{-\frac{1}{2}} \cdot (2\pi)^{-\frac{n}{2}} \cdot r_d. \quad (6)$$

6. Verification of the condition (von Neumann elimination),

$$R \leq f(\mathbf{x}|\mathbf{y}). \quad (7)$$

If this condition holds, vector \mathbf{x} is accepted and the simulation for the next time step is undertaken. If not, the calculation returns to point 3.

7. Determination of the mean generated vector, $\bar{\mathbf{x}}$, mean known vector, $\bar{\mathbf{y}}$, and the generated covariance matrix, $\bar{\mathbf{K}}$,

$$\bar{\mathbf{x}} = \frac{1}{NR} \sum_{i=1}^{NR} \mathbf{x}_i, \quad \bar{\mathbf{y}} = \frac{1}{NR} \sum_{i=1}^{NR} \mathbf{y}_i, \quad \bar{\mathbf{K}} = \frac{1}{NR-1} \sum_{i=1}^{NR} \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix} \right) \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix} \right)^T, \quad (8)$$

where NR is a number of time steps in the earthquake record.

8. Comparison between the assumed covariance matrix, \mathbf{K} , and the generated one, $\bar{\mathbf{K}}$, by calculation the global error,

$$GE = \left| \frac{\|\mathbf{K}\| - \|\bar{\mathbf{K}}\|}{\|\mathbf{K}\|} \right| \cdot 100\%, \quad (9)$$

where $\|\mathbf{K}\|$ and $\|\bar{\mathbf{K}}\|$ are the Euclidean norms for \mathbf{K} and $\bar{\mathbf{K}}$, respectively.

3 Numerical Example

In this paper, the example of numerical simulations of ground motion records at the corners of a 100 m long and 50 m wide building has been presented. The method has been used to generate time histories at 3 corners described by the coordinates: (0,50), (100,50) and (100,0) based on a specified earthquake record at (0,0). In the analysis, the generation of values at every time step is based on the known value at this time

step for a specified time history as well as the known values (previously generated) for a previous time step for all records. For the simulation purposes, the N-S component of the El Centro earthquake (May 18, 1940), with the time step of $\Delta t = 0.01$ s, has been used. The examples of the results of simulations obtained for $\nu = 1000$ m/s, $d = 1$ and $\beta = 1$ are shown in Fig. 1. The mean simulation error calculated according to Eq. (9) has reached the value of 1.78 %.

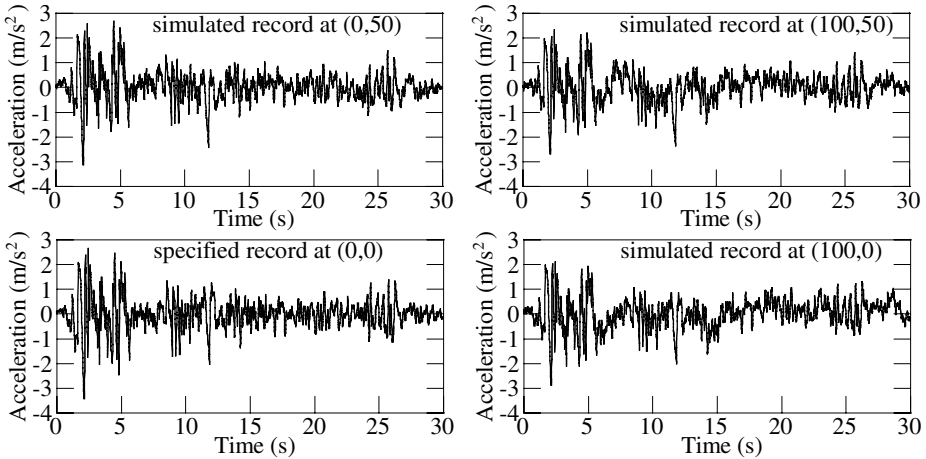


Fig. 1. Specified and simulated acceleration time histories for the El Centro earthquake

4 Concluding Remarks

The method of conditional stochastic simulation of ground motions using the spatio-temporal correlation function has been considered in this paper. The results of the study show that the method allows us to generate earthquake time histories with relatively low simulation errors and therefore can be suggested for practical applications.

References

1. Jankowski, R., Walukiewicz, H.: Modeling of two-dimensional random fields. *Probabilistic Engineering Mechanics* 12 (1997) 115-121
2. Jankowski, R., Wilde, K.: A simple method of conditional random field simulation of ground motions for long structures. *Engineering Structures* 22 (2000) 552-561
3. Wang, J., Carr, A., Cooke, N., Moss, P.: Effects of spatial variation of seismic inputs on bridge longitudinal response. *Proc. of 13th World Conf. on Earthquake Engineering*, Vancouver, Canada (2004) paper no. 640
4. Zerva, A., Zervas, V.: Spatial variation of seismic ground motions: An overview. *Applied Mechanics Reviews* 55 (2002) 271-297

A GIS Based Virtual Urban Simulation Environment

Jialiang Yao¹, Hissam Tawfik², and Terrence Fernando¹

¹ Future Workspaces Research Centre, University of Salford,
Salford, M5 4WT, UK

² Information Management & Communication, Liverpool Hope University College,
Liverpool, L16 9JD, UK

Abstract. This paper presents the development of a virtual reality urban planning tool based on GIS and Virtual Environments to facilitate scenario generation and collaborative urban design. Our system provides an open structure for users to support various visualisation and simulation modules, for the prototyping of urban designs. An Urban Scene Generation module is used within this urban planning system for generating the urban layout, on which the various collaborative planning tasks are carried out. A 3D urban model is generated from a GIS database by extracting features such as landscape, roads, and buildings. The system allows the users to place CAD models within the virtual environment. Various statistical data related to population, crime, and employment are also visualised in this virtual environment. A VR based user interface is provided to handle urban simulation data, query information and discuss urban simulation scenarios.

Keywords: Urban Planning, Virtual Reality, GIS.

1 Introduction

Urban planning is a complex and multidisciplinary process requiring the consideration of a number of economical, sociological, transportation and environmental issues. A successful urban planning project requires the satisfactory combination of cost, environmental and cultural requirements.

It is important at the early urban planning stage, to consider how a plan will affect the urban area [1]. Hence an urban plan, which is simulated and evaluated thoroughly at an early stage, would improve the urban regeneration project in terms of cost, efficiency and productivity. Therefore urban planning is a multi-objective decision making task which aims to plan the urban environment in terms of its physical, social, legal, economical, visual and environmental elements, and could achieve many benefits from systems that can help demonstrate and evaluate various aspects of urban planning at an early stage to key stakeholders and the local community. The adoption of a virtual prototyping design concept can bring decision makers and stakeholders together to exchange ideas, quickly assess alternative and build consensus.

Traditional urban planning uses basic tools, such as 2D-maps and text based documents, to deliver a satisfactory urban design solution to all the stakeholders. More recently, databases, computer graphics, photogrammetry and geographic information system (GIS) technology have been adopted to support the process of

urban planning [2]. Such tools have proved to be effective for professionals performing urban planning tasks. However, they still lack the capabilities for satisfying the increasing need for collaborative urban planning to allow a wide range of professionals involved in the process to exchange and discuss ideas and problems by visualising and interacting with complex data. The aim of our Collaborative Urban Planning Support System is to provide an intuitive and collaborative urban planning environment to enhance participation of stakeholders in urban planning through the use of innovative IT tools.

2 Related Work

Numerous information and communication technologies have been developed to aid the urban planning process in recent years, in response to the need for supporting urban planning through visualisation, urban simulation and public participation.

Chan et al. [3] presented an urban simulation system which combines VR, GIS and database technology, to establish a 3D urban simulation environment (Los Angeles Basin, USA). Bourdakis et al. [4] created a Bath model from 1992 using an industry standard CAD package, AutoCAD, that covers several square kilometres of the historic centre of Bath which is accurate to the sub-metre level. A considerable amount of realism in terms of geometrical representation has been modelled using geometry and texture mapping. City planners have used the model to test the visual impact of a number of proposed developments for the city's future. The Adelaide Model [5] demonstrated that the 3D GIS could provide an effective way to model the urban environment by extruding the third dimension from 2D maps. Although realistic 3D urban visualisation and simulation make the presentation of ideas and proposals at planning meetings more interesting and understandable for the public, the passive presentation of impressive "realistic" rendering is not enough for scenario simulation which is essential for collaborative urban planning and design [6]. Further research is still needed on the implications of realistic 3D urban models used in different stages of the negotiation and design process [7]. GIS is one of the commonly used methods in urban planning [5, 8], but the current systems only provide a way to visualise cities, and do not provide functions for manipulating the models, limiting their use in the collaborative design process.

Some research work also attempted to engage public participation in the planning process at every stage, such as the Woodberry Down Project [9]. It used VRML to deliver urban plans to the public in a visual form, but the feedback collected from the public was still in text form in bulletin boards, which may lose information [2] and cause misunderstandings during the collaborative planning procedure [10].

Very little research has been carried out to develop software environments to support urban planning using emerging technology such as virtual reality, 3D interaction and simulation. Addressing these issues requires the presentation of large and complex urban data types. U-Plan aims to facilitate more natural interaction with these datasets in a VR environment and provides an open architecture to facilitate scenario simulation and evaluation.

3 A Virtual Urban Planning Support Environment

3.1 Design Requirements

The aim of our VR Urban Planning Support System is to provide a virtual urban planning environment on which urban planners and stakeholders are able to visualise, manipulate and discuss urban planning issues in real time. This requires the system to have an intuitive interface for multi-user interaction and have the ability to quickly generate urban models and visualise various types of urban planning related datasets such as population, crime, employment and transport. The main user requirements for our system can be summarised as follows:

- **Urban models and urban planning scenario generation:** The system should have the ability to quickly generate an accurate, detailed and realistic 3D layout of an urban environment on which stakeholders can analyse urban planning issues.
- **Effective User Interface:** The system should have the ability to interact effectively within virtual environments to perform urban planning analysis, such as performing information query, object creation and removal in order to explore various options.
- **Collaborative capability:** Multi-user interaction within the virtual environment is needed to support the collaborative urban planning process.
- **Supporting different visualisation platforms:** The collaborative urban planning environment is required to run on different visualisation platforms, such as a standard workstation, workbench and Reality Room, in order to suit various decision-making and design prototyping purposes.

3.2 System Architecture

The proposed system architecture has a software framework that supports distributed interactive visualisation and simulation. The system utilises the Hydra distributed architecture, a microkernel for collaborative visualisation of dynamic simulations. In this environment, the servers are used for generating and managing the urban planning scenarios on which the various collaborative planning tasks and simulation data generation such as traffic network analysis can be carried out. A VR based user interface is provided to handle urban simulation data, query information and urban simulation scenarios. The user interface has been designed to run on a range of displays such as Workbench, Reality Room and standard workstation display (Figure 1).

Hydra is a software framework for distributed, multi-user interactive visualisation and simulation applications. It is designed around a lightweight core that assimilates plugins into the framework at runtime and manages the node layers and shared memory. An API layer enables application developers to interact with these data structures through a system of proxies. These proxies enable Hydra to manage the application requirements against the distribution of services across the network. No specific functionality is defined within the core. Instead, functionality is defined at runtime through the provision of user definable plugins, which play a variety of roles from functional adapters to generic service providers.

There are several visualisation clients for urban planning and simulation in the system. The client in the client-server architecture, typically, is an application that runs on a PC or workbench and relies on servers to perform some operations, as shown in Figure 2.

A World Manager module [11] is used to manage the Scene graph, viewer for scene graph visualisation and input layer for processing user inputs. The World Manager gets user input from input devices such as a head tracker and Wand which is a hand position tracker with programmable control buttons and passes corresponding data to the Viewer and Task Manager for visual feedback and input processing.

The 3D Widgets Module is a set of libraries, which provide a consistent interface from the desktop VR platform to a workbench. It is necessary to be able to interact with the system on a standard desktop system consisting of a monitor, keyboard, mouse and Spacemouse, to a workbench environment which consisting of head-tracker and wand. The Widget also provides cross dimensional support allowing tasks to be completed via 2D and 3D styles of interaction, along with gesture and voice based interaction. To enhance the usability of the system, a combination of WIMP techniques and 3D interaction techniques are supported by 3D Widgets.

This system architecture provides a consistent environment to support development of a distributed visualisation system such that different urban planners can share the same virtual urban scenario at different visualisation platforms and manipulate objects in this virtual urban environment. Further more, the framework provides an open architecture in which system functions can be defined at runtime through the provision of user definable modules.

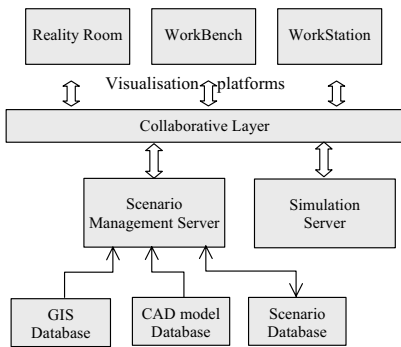


Fig. 1. The system architecture

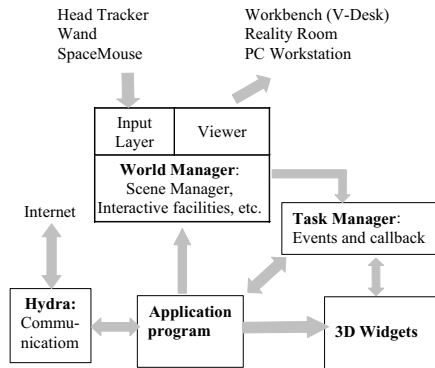


Fig. 2. Client Architecture

4 GIS Based Urban Model Generation

The generation of large-scale 3D urban models, using existing 3D modelling tools, such as AutoCAD and Relax, consumes a great deal of time and effort. In this system, ArcGIS is used to manage GIS database, and an urban generation model was developed using C++ and ArcObject to extract geometry and attribute features of the

landscape, roads, and buildings from ArcGIS database and generate 3D urban models for urban planning in virtual environments.

The 3D landscape model is generated from GIS TIN (Triangular Irregular Network) [12] Layer in which landscape geometry is defined using a group of 3D triangles. Aerial photos are used as texture data to increase the visual realism of the generated landscape.

The 3D Building model is generated according to the building’s footprint and height information in the GIS. The building’s footprint data is a 2D polygon geometry model [13]. The 3D building model is generated by extruding the polygon according to its height. The building base height is calculated based on its position on the landscape.

The 3D road model is generated using the road central line data from the GIS. A triangle strip generated from the road central line and its width is used to describe the 3D road. The height calculation of road is for the road’s triangle strip model, and the method to calculate the height of road is similar to that for buildings.

The 3D urban models use the same coordinate system as that used in GIS. In order to visualise urban models in virtual environments and carry out urban planning, a series of transforms are applied to urban models.

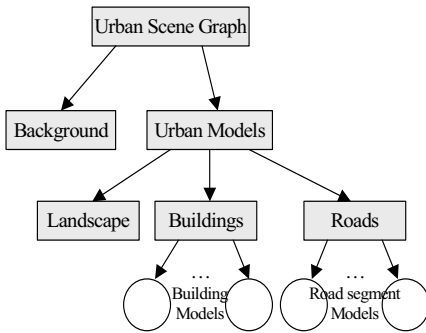


Fig. 3. Urban Scene Graph Organisation

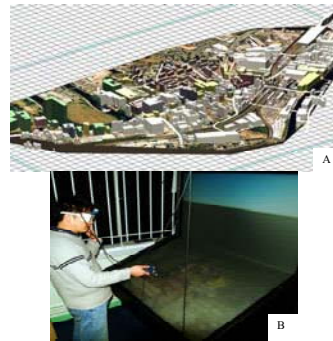


Fig. 4. (a) 3D Urban model and (b) urban model on WorkBench

A tree-like structural organisation method is used to manage and manipulate 3D objects in this system. The main objects include grids for sea-level ground reference, background objects for the surrounding environment and urban models which consists of the landscape, buildings and roads (Figure 3). The ID notations used for the 3D objects within the urban models are identical to their GIS database ID’s, making information query of the GIS system straightforward. Figure 4 shows the urban model of Chapel Street in Salford, UK.

The urban models generated from the GIS dataset can be modified through a user interface by adding extra models or removing existing models and saving them as different urban planning scenarios.

5 The User Interface

Based on collaborative urban planning process and a real urban regeneration project, we summarise the requirements for interaction within a virtual environment as follows:

1. Intuitive navigation is vital for collaborative urban planning. Users need the ability to explore any part in the virtual urban scene. This would help users to easily evaluate how a new developing plan affects the surrounding environment, such as how a new tall building affects the skyline of its surroundings and how a newly built supermarket affects the living or working of nearby residents.
2. The ability to create new objects, move and remove existing objects within a virtual urban environment to explore various design options.
3. The ability to mark or highlight an area within the urban environment to focus analysis and discussion around a particular area.
4. The ability to support the user to query information about objects within the urban environment, such as function of a building, highlight particular type of buildings in the area etc.

5.1 Implementation of Interaction Within Virtual Environments

Urban Planning scenarios are collaboratively presented on display platforms, where users are provided with series of functions to navigate, manipulate objects in virtual urban environment through interactive user interface.

The implementation of interaction facilities within the virtual environments is based on the world manager, the task manager and the widget manager which provide a consistent interface from the desktop VR to a immersive workbench. The main part of interaction includes system control, navigation and object selection and manipulation.

- System Control

In order to enable users to carry out necessary operations, the system uses a 3D Widgets Set [14] to provide a 3D form of WIMP-style user interface for the virtual environment. The 3D widgets set provide a series of tools, such as windows, menus (such as cascading menu and cube menu), buttons, text box and scroll list, and are used in virtual environments to facilitate operations such as information query, objects creation, object removal, and information display.

System interaction is conducted through the wand and SpaceMouse input devices. The wand, a tracked hand held device, is used for navigation, object selection, object manipulation and to execute system controls. The SpaceMouse is used for exocentric control of the 3D scene and perform same task as wand. In practice, the wand is more convenient in the Immersive Virtual Environment, while SpaceMouse is the tool used in the standard workstation.

The use of 3D widgets makes it possible to provide the user with a consistent interface between different platforms, and offers flexibility for users to perform information query, distance measuring, and buildings removal and creation.

• Navigation

An important aspect in developing the urban planning environment is to provide users with flexible navigational options to explore the urban environment and review urban development plan easily. Two navigational modes are used in this system, namely a world navigation mode and a target-based navigation mode. The World navigation mode, which uses birds-eye view method, is the default navigation mode that is used to allow users to navigate around the entire virtual space. Once an object is selected, the navigation mode will be changed to target based navigation mode (Figure 5), where the selected object will be kept in the centre of the screen.

A virtual compass is provided in this system to aid navigation and object manipulation. It is especially useful when users are unfamiliar with the virtual urban environment.

• Selection and manipulation

Object selection and manipulation are supported through direct interaction using manipulators. The Virtual Ray and virtual hand, which are controlled by input devices are used to conduct direct manipulation operations.

The Virtual Ray is used to select objects, measure distances, mark areas and manipulate widgets as most objects in virtual urban environment are out of reach of users' hands. In the default mode, once the virtual ray hits a selectable object, a bounding box of this object is displayed. If this object is selected using wand or SpaceMouse, the object's colour is changed and system changes its operation mode into Target Based Navigation. When measuring distances, the intersection point of the virtual ray with virtual urban models is traced. A polyline is used to show the path of measured place, and the measured distance is displayed on the toolbar (Figure 6).

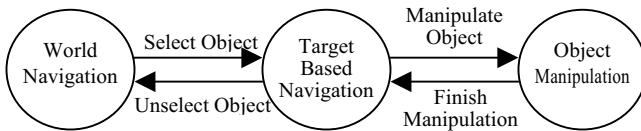


Fig. 5. Operation mode transition during

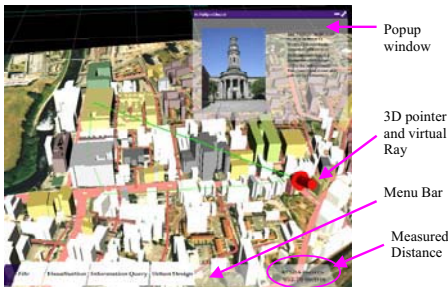


Fig. 6. Information query and distance measure- using virtual ray

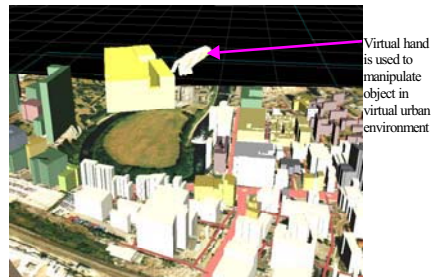


Fig. 7. Object creating, moving using virtual hand

The virtual hand is used to place objects, such as buildings, at a proper position. The hand's movement and orientation is used to control the object's movement and orientation. In the case of rotating buildings, only one parameter of the orientation – the rotation around Y-axis, is used to control the building's orientation. (Figure 7).

6 Summary

This work investigates the development of a Collaborative Urban Planning Support System that aims to enhance the urban planning procedure by enabling urban planners to collaboratively make, manipulate and evaluate urban development plans. Urban models are generated using GIS and displayed on Workbench, Reality Room and standard workstation platforms. An intuitive interface using 3D Widgets is provided for users to perform navigation, building manipulation, and information query operations. Initially user evaluation has shown that this system, especially when operating on the Workbench environment, provides an appropriate virtual urban planning platform.

Further development of this system will involve designing an analytical model for traffic network analysis and developing user interfaces suitable for multi-user operation.

Acknowledgement

This research forms part of the Virtual Urban Laboratory Computer Aided Network (VULCAN) project supported by the Salford local Council. The authors also acknowledge the contribution of S. Bee to the Hydra part and N. Murray to the 3D Widgets part of the system.

References

1. Jepson, W., Friedman, S.: A Real-Time Visualization System for Large Scale Urban Environments, White Paper (1998)
2. Goodfellow, D.: Collaborative Urban Design through Computer Simulations, A senior honours essay prepared for the School of Urban and Regional Planning, University of Waterloo, Ontario, Canada (1996)
3. Chan, R., Jepson, W., Friedman, S.: Urban Simulation: An Innovative Tool for Interactive Planning and Consensus Building. in *Proceedings of the 1998 American Planning Association National Conference*, Boston, MA, USA (1998) 43-50
4. Bourdakis, V.: From CAAD to VRML: London Case Study. in *The 3rd UK VRSIG Conference Full Paper Proceedings*, De Montfort University, UK (1996)
5. Kirby, S.D., Flint, R., Murakami, H., Bamford, E.: The Changing Role of GIS in Urban Planning: The Adelaide Model Case Study. *International Journal for Geomatics*, 11 (8) (1997) 6-8
6. Pietsch, S., Radford, A., Woodbury, R.: Making and Using a City Model, Adelaide, Australia. in *19th Conference of Education for Computer Aided Architectural Design in Europe*, Helsinki, Finland (2001) 442-447

7. Bourdakis, V.: Making Sense of the City. in *CAAD Futures 97*, (1997) 663-678
8. Batty, M., Dodge, M., Jiang, B.: GIS and Urban Design. *Geographical Information and Planning: European Perspectives* (1999) 43 - 65
9. Hudson-Smith, A., Evans, S.: Wired Regeneration: GIS in the third dimension *GIS@development* (2001)
10. Al-Kodmany, K.: Using Visualization Techniques for Enhancing Public Participation in Planning and Design: Process, Implementation, and Evaluation. *Landscape and Urban Planning*, 45 (1) (1999) 37-45
11. Marcelino, L., Murray, N., Fernando, T.: A User Interface for Virtual Maintainability in Immersive Environments. in *HCI International conference*, (2003)
12. Fowler, R.J., Little, J.J.: Automatic extraction of irregular network digital terrain models Automatic extraction of irregular network digital terrain models. *Computer Graphics*, 13 (2) (1979) 199-207
13. ESRI: *Modeling Our World, The ESRI Guide to Geodatabase Design*. Environmental Systems Research Institute, Inc. (1999)
14. Murray, N., Goulermas, J.Y., Fernando, T.: Visual Tracking for a Virtual Environment. in *HCI International conference*, (2003)

Scientific Workflow Infrastructure for Computational Chemistry on the Grid

Wibke Sudholt¹, Ilkay Altintas², and Kim Baldridge^{1,2}

¹ Institute of Organic Chemistry, University of Zurich, Winterthurerstrasse 190,
CH-8057 Zurich, Switzerland
{wibke, kimb}@oci.unizh.ch

² San Diego Supercomputer Center (SDSC), UC San Diego, 9500 Gilman Drive,
La Jolla, CA 92093-0505, USA
altintas@sdsc.edu

Abstract. We present ongoing research in the Resurgence (RESearch sURGe ENabled by CyberinfrastructurE) project. This infrastructure shall enable the flexible combination of computational chemistry tools from a unified interface, with the focus on automated high-throughput processing. The implementation is based on the idea that the time-consuming parts of the calculations can be distributed onto computational Grids using the Kepler scientific workflow system and the Nimrod toolkit for distributed parametric modeling. We describe an example workflow that allows preparing, running, and displaying jobs on different molecules, employing the GAMESS quantum chemical program package.

1 Introduction

1.1 Scientific Background

Catalyzed by life and material sciences, studying vast amounts of molecules becomes more and more important in applied chemical research. In the drug discovery pipelines of pharmaceutical companies, many thousand compounds are screened for activity in a high-throughput manner, often using computational chemistry approaches such as QSAR (Quantitative Structure Activity Relationships) and docking. Fundamental chemistry research, on the other hand, has traditionally been sparsely automated. However, established computational chemistry tools such as quantum chemistry, molecular dynamics, and continuum electrostatics nowadays allow the accurate prediction of molecular properties. Advances in technology continuously speed up these calculations. Together with the opportunities of data mining, one hopes to learn new chemical concepts from analyzing many molecular computations.

Moreover, driven by the interest in large and complex systems such as biomolecules, solutions, and materials, it is increasingly necessary to couple different computational chemistry packages. Algorithms from various scientific domains are combined, and resolutions range over several time and length scales, an important example being the QM/MM (Quantum Mechanics/Molecular Mechanics) techniques. However, these studies usually require the direct integration of application codes. The

purpose of the Resurgence project is thus the flexible and reusable coupling of computational chemistry packages. This will allow easier exchange of data between applications to realize new scientific ideas, and building of pipelines for automation of tasks. Our focus is on calculations for different compounds or conformers.

1.2 Distributed Scientific Computing

The underlying physical equations often make computational chemistry calculations very compute- and data-intensive. Complexity of the parallelized codes often results in highly communication-dependent algorithms. However, technology moves to loosely-coupled Grid infrastructures. Computational Grids are defined as “a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities” [1]. Virtual organizations manage the contribution and distribution of resources on a Grid. Established Grid projects have created new opportunities for computational research.

Unfortunately, apart from a few examples (e.g., [2, 3, 4]), building and using Cyberinfrastructure for computational chemistry has been lagging behind the hype. Not many applications have been ported to a Grid infrastructure, exceptions being for example the OpenMolGRID [5] and Gemstone [6]. Due to the long runtimes and high resource demands of the calculations, service-oriented architectures are seldom applied. Computational chemistry programs often result from years of collaboration. To gain speed, they are mainly written in procedural programming languages such as Fortran or C, using formatted input and output file formats specific to each code. Many programs are commercial, and only available as binaries or in-house. This makes it difficult to adapt computational chemistry packages to the requirements of Grid computing. But also Grid computing itself has partly stayed behind its promises. Most Grid middleware has long been in development state and exposes APIs complex to work with for non-specialists. Many management and policy issues have to be solved before a Grid infrastructure can be set up, which is often only accessible for selected researchers. Finally, network latency over long distances, hardware and software heterogeneity, and missing control on resource availability in Grids make it less efficient to port the highly-coupled parallel computational chemistry programs.

Nevertheless, Grid computing has a lot to offer for computational chemistry. Neighboring disciplines such as bioinformatics and high-energy physics with similar problems, though stronger orientation towards web services and data Grids, respectively, have demonstrated the benefits of using such infrastructures. Embarrassingly parallel calculations such as parameter sweeps and optimizations are perfectly suited for running on Grids. Corresponding tools such as Nimrod [7] and APST [8] provide wrappers around existing codes to distribute them over a number of machines, and have been successfully applied in scientific studies. What these tools are missing though are simple ways to seamlessly couple different applications to form pipelines, where the results of one step are used as basis for the next. The purpose of the Resurgence project is thus to provide a workflow tool that allows to flexibly combine computational chemistry codes within one common interface. This should be easy to use for beginners, hiding the complexity of the middleware, but details should be accessible for advanced users. Compute-intensive steps should be distributed onto the Grid, while short and interactive steps should stay on the local machine.

1.3 Scientific Workflow Systems

As the scientific process moves into an era of distributed data and computation, the needs of scientific studies have also changed. Now, the solution to a scientific problem often involves reaching out to Grids, discovering datasets, running pipelined and repeated analysis tasks in parameterized calculations, storing the outcomes into databases, registering the workflows and results into online portals, and visually displaying them. Today's scientists are expected to use all these technologies.

Scientific workflows emerged as a glue to compose different technologies by providing interfaces to run a coordinated set of data and computation components in a systematic and automated way. They can be applied to combine data integration, analysis, and visualization steps into larger 'knowledge discovery pipelines' and 'Grid workflows' [9]. Scientific workflow research and system development have recently been the focus of several big efforts. However, the lack of generic graphical interfaces that abstract the technical details from users and let them focus on the science, has been a major obstacle for using traditional workflow systems rather than customized applications. Scientific workflow user environments, for example Kepler [10], Taverna [11], and Triana [12], aim at improving this situation by 'wrapping' Grid tools and making them available in a user-friendly visual programming environment.

Kepler is particularly suited for computational chemistry workflows, as it does not only provide web service, Grid, and XML-based features, but also command-line wrapping, file system interaction, and string processing capabilities. The rest of this paper describes our case study in using Kepler for a computational chemistry pipeline. We first explain the requirements of the scientific problem and how different tools were used together to solve it. Then we describe how we utilized Kepler's components to assemble the workflow. Finally, we discuss the results and unresolved issues.

2 Tools

2.1 Computational Chemistry Programs

In the first Resurgence [13] implementation, we concentrate on the preparation, execution, and analysis of quantum chemical calculations on a number of small molecules. Our main principles are the restriction to open source codes to retain the accessibility for all users, and the use of downloadable versions of all application programs without modifications. The core tool employed is the GAMESS ab initio quantum chemistry package [14]. It is a well-established open source code to determine chemical properties based on the Schroedinger equation. It is mainly written in Fortran, highly parallelized, and can run very efficiently on different computer architectures. As example of an analysis tool for the calculation results we selected the molecular graphics program QMView [15]. It allows displaying and modifying the three-dimensional structures of chemical compounds and provides a large variety of options to examine their molecular properties. An important issue in computational chemistry studies is related to file formats and their transformation. Despite emerging standards such CML (Chemical Markup Language) [16], the vast majority of applications still use their own legacy input and output file formats. This is one of the main limiting

factors of program interoperability. Therefore, we included two key file format conversion tools, Open Babel [17] and its predecessor Babel [18].

2.2 Nimrod Distributed Parametric Modeling Toolkit

As Grid computing interface for the parameterized calculation jobs, Nimrod/G [7] is used. It is part of the Nimrod toolkit for distributed parametric modeling and was already successfully employed previously [2]. Nimrod can perform embarrassingly parallel parameter sweeps (Nimrod/G) and optimizations (Nimrod/O) distributed on a Computational Grid. The user specifies a list of available computer resources, and for each computational ‘experiment’ (i.e., collection of parallel jobs), selects the particular resources to use. A short, declarative ‘plan file’ describes the parameters to consider and the tasks to be performed for each single job. Nimrod then allocates, executes, controls, and collects the individual jobs via ‘agents’ submitted automatically to the involved resources. Nimrod only has to be installed once on the central root node, and then communicates with the Grid nodes via the corresponding middleware, mainly the Globus toolkit [19]. At the backend of Nimrod is a PostgreSQL database and Python code. It has a command-line interface, which we employ here, a web portal (Nimrod Portal), and an Excel interface (Active Sheets).

2.3 Kepler Workflow Approach

We have selected Kepler [10] as the environment to utilize the above-mentioned tools in a workflow. Kepler is a collaboration between several domain research projects, to develop an extensible and customizable open-source scientific workflow system. The Kepler system allows scientists from different disciplines (bioinformatics, cheminformatics, ecoinformatics, geoinformatics, oceanography, astrophysics, etc.) to design and execute scientific workflows. It builds on top of the mature Ptolemy II software developed at UC Berkley [20]. Ptolemy II is a Java-based system along with a set of APIs for heterogeneous hierarchical modeling. The focus of Ptolemy II is to build models based on the composition of existing components, which are called ‘actors’. Actors are encapsulations of parameterized actions performed on input data to produce output data. Inputs and outputs are communicated through ports within the actors. The interaction between the actors is defined by Models of Computation (MoC). A MoC specifies the communication semantics among ports and the flow of control and data among actors. Directors are responsible for implementing particular MoCs, and thus define the ‘orchestration semantics’ for workflows. By changing the director, one can change the scheduling and execution semantics of a workflow, without changing any of the components or the network topology. An example for a director is the ‘Synchronous Data Flow (SDF)’ domain, in which a sequential execution order of actors can be statically determined prior to execution. Here we need to combine it with the more dynamic ‘process networks (PN)’ director, which also demonstrates the usage of multiple execution domains in one workflow. The descriptions of Kepler workflows are saved in an XML-based format called Modeling Markup Language (MoML). MoML is the primary persistent file format for Ptolemy II models and Kepler workflows. It also represents the primary mechanism for constructing workflows whose definition and execution is distributed on the network.

4 Implementation and Results

Our initial case study tests the capabilities of Kepler, and the applicability of Kepler workflow development methodologies to computational chemistry calculations and the spreading of tasks onto distributed nodes. The conceptual workflow underlying our example implementation is simple (see Figure 1). Several files containing the atomic coordinates of different molecules are provided. From these, input files for the quantum chemical program package GAMESS are generated. Then the corresponding calculations are executed. Finally, the results of the computations are visualized using the QMView software. A pipeline like this is part of the daily work of computational chemists. The largest variant is usually that different molecular systems, application codes, and keyword and parameter settings are used for the preparation, calculation and visualization steps. If more than a few compounds have to be processed, the typical practice is to use shell scripts. The Resurgence implementation into Kepler provides an easier, more flexible, and reusable manner to accomplish the same tasks.

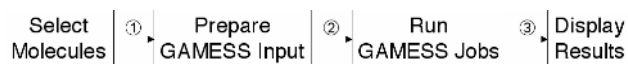


Fig. 1. Conceptual workflow

However, it is clear that the workflow in Figure 1 cannot be directly realized, as different file formats are involved. In our example, we start with CML files, which are then transformed into GAMESS input files, processed into GAMESS output files, and finally converted into PDB (Protein Data Bank) [21] files, which can be displayed. Other format transformations would be possible. This means that at steps number 1 and 3, additional format conversion programs have to be included, in our case Open Babel and Babel. Moreover, the GAMESS calculations are very resource-intensive and time-consuming, and thus the most interesting step to target for Grid distribution. On the other hand, the graphical display and the data-intensive, but fast file format transformations are currently better performed on the local machine. However, if management and storage requirements for these steps increase in future applications, database or data Grid capabilities already available in Kepler might be employed.

The major Kepler actors used for the construction of the executable workflow for the mentioned conceptual workflow are the *CommandLine* and *ExternalExecution* actors as well as generic textual display and various constant, array, string function, and file system manipulation actors. The *CommandLine* and *ExternalExecution* actors are wrappers for executing commands that run legacy code in a workflow. Their purpose here is to call the diverse employed external programs or corresponding automatically generated temporary shell-script wrappers. To conceptually organize the workflow and to be as modular as possible, we have extensively used composite actors and MoML actor classes. Composite actors are combinations of other actors that are implemented as a block diagram and can be saved into the actor library for reusing and updating purposes. Actor classes allow applying these composite entities in workflows in a similar way as one can use native atomic actor Java classes.

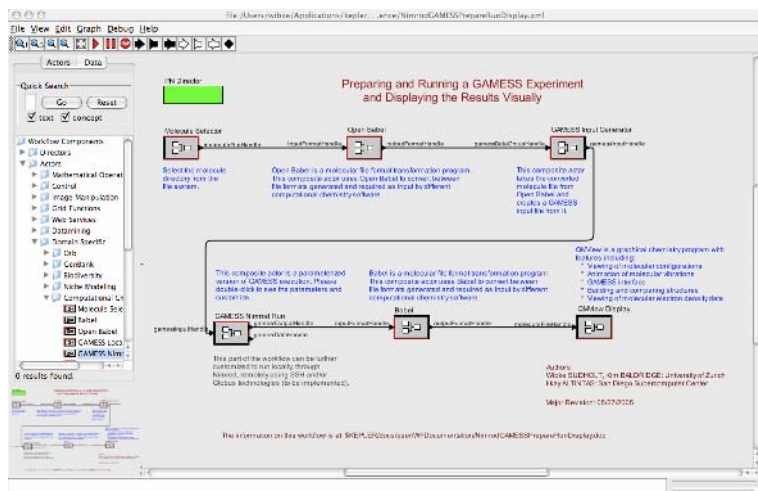


Fig. 2. GAMESS/Nimrod workflow in Kepler

Figure 2 illustrates the Kepler view for the workflow that implements the conceptual workflow in Figure 1. The constructs that make up the workflow were abstracted to six main steps: ‘Molecule Selector’ to choose the input files for a set of target molecules, ‘Open Babel’ to transform these files into the GAMESS input format, ‘GAMESS Input Generator’ to annotate the inputs with the right set of calculation keywords, ‘GAMESS Nimrod Run’ for the Grid execution, ‘Babel’ for transformation of GAMESS outputs to PDB format, and finally, ‘QMView Display’ to visualize the outputs for each molecule. The only data sent between these actors are sequences of file handles belonging to the individual molecules. The corresponding file name bases also serve as the ‘parameter’ in the auto-generated intermediate Nimrod/G plan file. On this highest-level view of the workflow, the most important variables of the computations can be specified. However, all described steps are composite actor classes which have several layers of sub-workflows hidden inside. By opening and modifying them, the user can fine-tune the calculation pipeline to various degrees. We have tested the workflow for geometry optimizations of a number of different small molecules on a Mac laptop for local execution. For distributed Nimrod execution, a small Globus-based Grid consisting of two seven and eight-node Xeon Linux clusters can be used. Results from ongoing tests will be made available online.

5 Discussion and Conclusions

The presented example workflow allows to easily setup a pipeline to prepare, run, and display quantum chemical calculations on different compounds. The compute-intensive part may be distributed onto a computational Grid with the help of the Nimrod toolkit, reducing the time for this embarrassingly parallel step. Due to the graphical visualization of the results at the end, the number of molecules that can be treated concurrently is limited. The most important parameters, such as file types, directory names, external binary locations and arguments, important GAMESS keywords,

as well as the experiment name, can be specified on the highest workflow level. Tuning of further details of the calculations, for example seldom used GAMESS keywords and Grid node set up can be modified using sub-components of composite actor classes. Due to its high modularity, each component can be reused in other workflows. The GAMESS Nimrod execution actor can substituted by a GAMESS local execution actor without changing the rest of the pipeline. Overall, our implementation is expected to enhance the design and execution of computational chemistry pipelines. By reducing complexity and increasing manageability, adaptability, and reusability as well as by transparent access to Grid resources, preparation and calculation time and effort decreases and more advanced workflows are possible. However, success will finally be measured by its usage scientific studies.

Despite this progress, we did not reach all our goals yet. The integration of the external application codes GAMESS, QMView, Open Babel, and Babel is already far developed. However, the implementation of Nimrod/G currently only covers a part of its functionality. In addition to Nimrod and local execution, implementing SSH, Globus, and web service capabilities for GAMESS is planned. Certainly, the Resurgence interface should be expanded to further domain-specific computation codes. However, also some fundamental aspects of our solution need to be looked at. Its robustness in real high-throughput settings remains to be tested. The necessary file format transformations could benefit from a systematic and rigorous typing system. And finally, the logging, monitoring, and control of the long-running computational experiments have to be improved. The Resurgence module of the Kepler scientific workflow system can be downloaded free of charge from the Kepler website and CVS.

Acknowledgements

We thank Celine Amoreira and Yohann Potier from the University of Zurich, Adam Birnbaum and Jerry Greenberg from SDSC, Yang Zhao from UC Berkeley, and David Abramson and his group from Monash University, Australia, for collaboration. W.S. acknowledges support from the CoLab at ETH Zurich, K.B. from the NSF-NMI (ANI-0223043), NSF-PRAGMA, and NIH-NCRR (NBCR-RR08605).

References

- [1] Foster, I., Kesselman, C.: Computational Grids. In I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufman(1999), pp. 15-51.
- [2] Sudholt, W., Baldrige, K.K., Abramson, D., Enticott, C., Garic, S., Kondric, C., Nguyen, D.: Application of grid computing to parameter sweeps and optimizations in molecular modeling. *Future Generation Computer Systems*, 21 (2005), pp. 27-35.
- [3] Baldrige, K.K., Greenberg, J.P., Sudholt, W., Mock, S., Altintas, I., Amoreira, C., Potier, Y., Birnbaum, A., Bhatia, K., Taufer, M.: The Computational Chemistry Prototyping Environment, *Proceedings of the IEEE*, 93 (2005), pp. 510-521;
- [4] Baldrige, K.K., Sudholt, W., Greenberg, J.P., Amoreira, C., Potier, Y., Altintas, I., Birnbaum, A., Abramson, D., Enticott, C., Garic, S.: Cluster and Grid Infrastructure for Computational Chemistry and Biochemistry. In A.Y. Zomaya, ed., *Parallel Computing for Bioinformatics and Computational Biology*, Wiley (2005), pp. 533-552.

- [5] <http://www.openmolgrid.org/>.
- [6] <http://www.baldridge.unizh.ch/gemstone/>.
- [7] Abramson, D., Giddy, J., Kotler, L.: High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?. International Parallel and Distributed Processing Symposium (IPDPS), Cancun, Mexico (2000), pp. 520-528; <http://www.csse.monash.edu.au/~davida/nimrod/>.
- [8] <http://grail.sdsc.edu/projects/apst/>.
- [9] Altintas, I., Birnbaum, A., Baldridge, K.K., Sudholt, W., Miller, M., Amoreira, C., Potier, Y., Ludaescher, B.: A Framework for the Design and Reuse of Grid Workflows. Scientific Applications of Grid Computing: First International Workshop, SAG 2004, Beijing, China, LNCS, 3458 (2004), pp. 119-132.
- [10] Ludaescher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows (2005), to appear; <http://www.kepler-project.org/>.
- [11] <http://taverna.sourceforge.net/>.
- [12] <http://www.trianacode.org/>.
- [13] <http://www.baldridge.unizh.ch/resurgence/>.
- [14] Schmidt, M.W., Baldridge, K.K., Boatz, J.A., Elbert, S.T., Gordon, M.S., Jensen, J.H., Koeski, S., Matsunaga, N., Nguyen, K.A., Su, S.J., Windus, T.L., Dupuis, M., Montgomery, J.A.: The General Atomic and Molecular Electronic Structure System. J. Comput. Chem., 14 (1993), pp. 1347-1363; <http://www.msg.ameslab.gov/GAMESS/>.
- [15] Baldridge, K.K., Greenberg, J.P.: QMView: A computational chemistry three-dimensional visualization tool at the interface between molecules and mankind. J. Mol. Graphics, 13 (1995), pp. 63-66; <http://www.nbcr.net/software/QMView/>.
- [16] <http://www.xml-cml.org/>; <http://cml.sourceforge.net/>.
- [17] <http://openbabel.sourceforge.net/>.
- [18] <http://www.ccl.net/cca/software/UNIX/babel/>.
- [19] <http://www.globus.org/>.
- [20] <http://ptolemy.eecs.berkeley.edu/ptolemyII/>.
- [21] <http://www.rcsb.org/pdb/>.

Application of the Reactivity Index to Propose Intra and Intermolecular Reactivity in Catalytic Materials

Abhijit Chatterjee

Material Science, Accelrys, Nishishinbashi TS Bldg. 11F, 3-3-1 Nishishinbashi,
Minato-ku, Tokyo, 105-0003, Japan
achatterjee@accelrys.com

Abstract. This study is based on our earlier work with reactivity index to propose intra and inter molecular reactivity in catalytic materials using density functional theory, within the domain of hard soft acid base (HSAB) principle. We have as well shown small example to show directly the utility of this method in elucidating acidity in catalytic material of interest. Our goal is to show that a simple theory can be useful to design new futuristic material of interest.

1 Introduction

The hard soft acid-base (HSAB) principles classify the interaction between acids and bases in terms of global softness. Pearson proposed the global HSAB principle [1]. The global hardness was defined as the second derivative of energy with respect to the number of electrons at constant temperature and external potential, which includes the nuclear field. The global softness is the inverse of this. Pearson also suggested a principle of maximum hardness (PMH) [2], which states that, for a constant external potential, the system with the maximum global hardness is most stable. In recent days, DFT has gained widespread use in quantum chemistry [3]. This study aims to review the development and application of reactivity index in key catalytic process within our research [4-9].

With this background, the influence of both bivalent and trivalent metal substituents from a range of metal cation (Co, Mn, Mg, Fe and Cr) on the acidic property (both Bronsted and Lewis) of metal substituted aluminum phosphate MeAlPOs is monitored. Intramolecular and intermolecular interactions show that once active site of the interacting species is identified, the influence of the environment can be prescribed.

2 Theory

The electronegativity χ , defined by Mulliken as the average of ionization potential I and electron affinity A , $\chi = 1/2(I + A)$, is such a parameter; it is a useful measure of the tendency of a species to attract electrons. Thus the electronegativity χ has been identified as the negative of the chemical potential μ , which is the Lagrange multiplier in the Euler-Lagrange equation in density functional theory, and

$$\mu = (\delta E / \delta N) v(r) = -\chi \quad (1)$$

Where, E is the total electronic energy, N is the number of electrons, and $v(r)$ is the external electrostatic potential an electron at r feels due to the nuclei. Mulliken's formula - $1/2(I + A)$ is no more than the finite-difference approximation for Eq. 1.

Similarly, the natural definition of the hardness, called η has been shown to be

$$\eta = (\delta^2 E / \delta N^2) v(r) = (\mu \delta / \delta N) v(r) \quad (2)$$

Which, has the finite-difference approximation $I - A$. The hardness/softness/acid/base principle has been derived using Eq. 2 as the definition of hardness. On the other hand, the frontier electron densities proposed by Fukui are local properties that depend on r ; they differentiate one part of a molecule from another and serve as reactivity indices.

In density functional theory, hardness (η) is defined as

$$\eta = 1/2(\delta^2 E / \delta N^2) v(r) = 1/2 (\delta \mu / \delta N)_v \quad (3)$$

Where E is the total energy, N is the number of electrons of the chemical species and the chemical potential.

The global softness, S , is defined as the inverse of the global hardness, η .

$$S = 1/2\eta = (\delta N / \delta \mu)_v \quad (4)$$

Using the finite difference approximation, S can be approximated as

$$S = 1/(IE - EA) \quad (5)$$

where IE and EA are the first ionization energy and electron affinity of the molecule, respectively.

The Fukui function $f(r)$ is defined by

$$f(r) = [\delta \mu / \delta v(r)]_N = [\delta \rho(r) / \delta N]_v \quad (6)$$

The function 'f' is thus a local quantity, which has different values at different points in the species, N is the total number of electrons, μ is the chemical potential and v is the potential acting on an electron due to all nuclei present.

The local softness $s(r)$ can be defined as

$$s(r) = (\delta \rho(r) / \delta \mu)_v \quad (8)$$

Equation (9) can also be written as

$$s(r) = [\delta \rho(r) / \delta N]_v [\delta N / \delta \mu]_v = f(r)S \quad (7)$$

Thus, local softness contains the same information as the Fukui function $f(r)$ plus additional information about the total molecular softness, which is related to the global reactivity with respect to a reaction partner, as stated in HSAB principle. Atomic softness values can easily be calculated by using eq 10, namely:

$$s_x^+ = [q_x(N+1) - q_x(N)]S \quad (8)$$

$$s_x^- = [q_x(N) - q_x(N-1)]S$$

$$s_x^0 = S[q_x(N+1) - q_x(N-1)]/2$$

We further explained the interaction energy scheme as follows. This is known that A & B interacts in two steps: (1) interaction will take place through the equalization of chemical potential at constant external potential and (2) A and B approach the equilibrium state through changes in the electron density of global system generated by making changes in the external potential at constant chemical potential. That means within DFT we can write

$$\Delta E_{\text{inter}} = E[\rho_{AB}] - E[\rho_A] - E[\rho_B] \quad (9)$$

Where ρ_{AB} , ρ_A , ρ_B are the electron densities of the systems AB at equilibrium and of the isolated systems A and B, respectively

Or in terms of the potentials we can write

$$\Delta E_{\text{inter}} = \Delta E_v + \Delta E_\mu \quad (10)$$

Where, $\Delta E_v = -1/2[(\mu_A - \mu_B)^2 / (S_A + S_B)] (S_A S_B)$

$$\Delta E_\mu = -1/2 N_{AB}^2 k [1/(S_A + S_B)] \quad (11)$$

N_{AB} total no. of electrons, k the proportionality constant between S_{AB} and $S_A + S_B$, product of N^2 and K is λ .

$$\Delta E_\mu = (-1/2) \lambda / (S_A + S_B) \quad (12)$$

If the interaction is taking place through j site of A,

$$\lambda_{Aj} = q_{Aj}^{\text{eq}} - q_{Aj}^0 \quad (13)$$

q_{Aj}^{eq} is density of jth atom of A in complex AB & q_{Aj}^0 is the density in isolated system.

In our all studies, all calculations with molecular clusters have been carried out with DFT [11] using DMol³ code of Accelrys. A gradient corrected functional BLYP [12] and DNP basis set [13] was used through out the calculation. Single point calculations of the clusters in their cationic and anionic form, at the optimized geometry of the original neutral clusters were also carried out to evaluate Fukui functions and global and local softness. The condensed Fukui function and atomic softness were evaluated using eqs 8 and 11, respectively, using electrostatic potential (ESP) driven charges.

3 Result

We will now site a short example to show how these calculations can be implemented to design catalytic materials with acidic site activity. Calculations were performed on AIPO34 framework, which is isostructural with zeolite chabazite. The cluster calculations were formed on localized cluster generated from the AIPO34 structure with the terminal Al or P. Two independent clusters of the formula (1) $M^{+2}AlP_2O_{12}H_9$ and (2) $M^{+3}AlP_2O_{12}H_8$ generated by replacing one P by a M^{+2} or M^{+3} to represent the bivalent and trivalent dopant incorporated clusters respectively as shown in Figure 1. The proton is included at the bridging oxygen where the dopant is incorporated for electrical neutrality for bivalent substitution. The terminal Al or P were replaced by hydrogen at that distance to mimic the real situation. It is observed that for the bivalent dopants the local environment is a distorted tetrahedral. For all the cases the M-OH distance is the longest. The M-O distances are considerably longer than the Al-O distance when the AIPO material is undoped, showing that the dopants introduce a considerable amount of distortion in the system. There is a drastic change in the M-O-P and angle values, $\sim 135^\circ$ compared to the Al-O-P $\sim 148^\circ$, which show that the observed structural distortion is not local and can be propagated beyond the nearest neighbor to the undoped region, which is in sharp contrast to the earlier results of Saadoune et al [14].

To correlate the activity of dopants, hence we performed localized reactivity index calculation for the bivalent dopants using $M^{+2}AlP_2O_{12}H_9$ cluster. The Fukui function and local softness for the hydroxyl proton is presented both in terms of nucleophilic and electrophilic activity. Relative electrophilicity (s_x^+ / s_x^-) and relative nucleophilicity

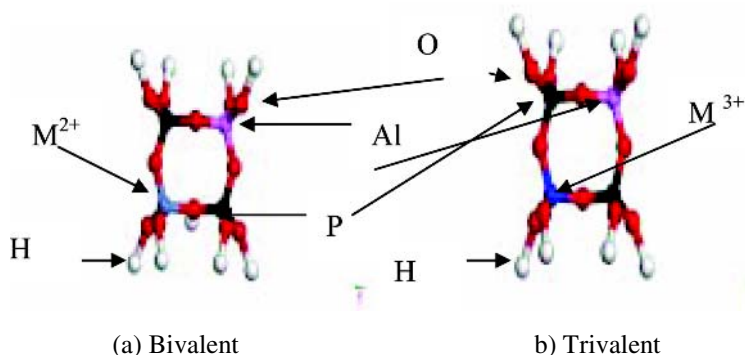


Fig. 1. Two independent cluster with the formula (a) $M^{2+}AlP_2O_{12}H_9$ and (b) $M^{3+}AlP_2O_{12}H_8$ to represent the bivalent and trivalent dopant incorporated clusters. Color code is as follows: red oxygen, black phosphorous, light pink aluminum, light blue is the bivalent, deep blue is trivalent metal cation, white is the hydrogen.

Table 1. Local softness and relative nucleophilicity for the bivalent dopants calculated in terms of the hydroxyl proton using ESP charges by DFT to monitor Bronsted acidity trend

Metal ion	s_x^+	s_x^-	s_x^+/s_x^-
Mg ²⁺	0.20	0.53	2.555
Mn ²⁺	0.37	0.32	1.156
Cr ²⁺	0.43	0.44	0.977
Co ²⁺	0.53	0.36	1.478
Fe ²⁺	0.53	0.44	1.204

(s_x^-/s_x^+) can be defined as the electrophilicity of any site as compared to its own nucleophilicity for the first term and vice versa. Table 1. Local softness and relative nucleophilicity for the bivalent and trivalent dopants calculated in terms of the dopant using ESP charges by DFT to monitor Lewis acidity trend.

The relative nucleophilicity is highest for Mg²⁺ and is lowest for Cr²⁺, which is opposite to the trend observed in terms of substitution energy. Fukui functions were used to monitor the dopants activity in terms of Lewis acidity. The results are shown in Table 2. For bivalent cation this order is totally different from the order obtained in terms of substitution energy and that obtained from the Bronsted acidity trend. For the trivalent dopant the highest nucleophilicity lies with Fe³⁺ and the lowest is for Mn³⁺; the results matches with the trend of substitution energy. The trend for Bronsted acidity is Cr²⁺ < Mn²⁺ < Fe²⁺ < Co²⁺ < Mg²⁺, where as the trend for Lewis acidity mainly for trivalent metal dopant is Fe³⁺ > Co³⁺ > Cr³⁺ > Mn³⁺. This optimistic result encourages us to monitor a mixed valence situation, which may ideally exist during calcinations for the cations with variable oxidation state.

4 Conclusion

The current work with all our previous work described here is aimed to resolve the issues within the domain of catalysis, we are now capable of using reactivity index to design new catalytic material of interest. The theory derived has ample application in analytical chemistry, industrial chemistry, organic chemistry and all. This is simple tool based on basic theory of donor and acceptor behavior with a great impact to resolve correlated issues.

Reference

1. Pearson, R. G. Recent advances in the concept of hard and soft acids and bases. *J. Chem. Edu.* 64 (1987) 561-567.
2. Parr, R. G., and Pearson, R. G. Absolute hardness - companion parameter to absolute electronegativity. *J. Am. Chem. Soc.* 105 (1983) 7512-7516.
3. Geerlings, P., and De Proft, F. HSAB principle: Applications of its global and local forms in organic chemistry. *Int. J. Quant. Chem.* 80 (2000) 227-235.
4. Chatterjee, A., Iwasaki, T., and Ebina, T. Reactivity index scale for interaction of heteroatomic molecules with zeolite framework. *J. Phys. Chem. A.* 103 (1999) 2489-2494.
5. Chatterjee, A. and Iwasaki, T. A Reactivity Index Study to Choose the Best Template for a Particular Zeolite Synthesis, *J. Phys. Chem. A* 105 (2001) 6187-6196.
6. Chatterjee, A., Iwasaki, T. and Ebina, T. Best dioctahedral smectite for nitrogen heterocyclics adsorption – a reactivity index study, *J. Phys. Chem. A* 105 (2001) 10694-10701.
7. Chatterjee, A., Ebina, T., Iwasaki, T. and Mizukami, F. Intermolecular Reactivity Study to Scale Adsorption Property of Para and Meta Substituted Nitrobenzene over 2:1 Dioctahedral Smectite *J. Chem. Phys.* 118 (2003) 10212-10220.
8. Chatterjee, A., Ebina, T., Onodera, Y. and Mizukami, F. 2,3,7,8 tetrachloro Dibenzo-p-dioxin Can Be Successfully Decomposed over 2:1 Dioctahedral Smectite – A Reactivity Index Study, *J. Mol. Graphics & Modeling* 22 (2003) 93-104.
9. Chatterjee, A., Suzuki, T.; Takahashi, Y.; Tanaka, D.A.P. A density functional study to choose the best fluorophore for PET sensor *Chemistry – A European Journal* 9 (2003) 3920-3929.
10. Chatterjee, A., Ebina, T., Onodera, Y. and Mizukami, F. Effect of exchangeable cation on the swelling property of 2:1 dioctahedral smectite – a periodic first principle study, *J. Chem. Phys.* 120 (2004) 3414-3424.
11. Delley, B., An all-electron numerical-method for solving the local density functional for polyatomic-molecules. *J. Chem. Phys.* 92 (1990) 508-517.
12. Becke, A. D., A multicenter numerical-integration scheme for polyatomic-molecules. *J. Chem. Phys.* 88 (1988) 2547-2553.
13. Lee, C. T., Yang, W. T., and Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron-density. *Physical Review B.* 37 (1988) 785-789.
14. Saadoune, I.; Catlow, C.R.A.; Corà, F. Site ordering of dopant ions in microporous aluminophosphates—size effects *Microporous and Mesoporous Materials*, 59 (2003) 161-165.

Conformational Processes in L-Alanine Studied Using Dual Space Analysis

Chantal T. Falzon and Feng Wang

Centre for Molecular Simulation, Swinburne University of Technology, P.O. Box 218,
Hawthorn, Melbourne, Victoria, Australia, 3122
fwang@swin.edu.au

Abstract. Binding energy spectra and orbital momentum distributions of the two most stable conformers of L-alanine are investigated. Molecular properties such as geometry and dipole moments agree well with available experimental and previous theoretical investigations. Dual space analysis is employed to study the binding energy spectra in coordinate space based on B3LYP/TZVP density functional calculations, and the valence orbital momentum distributions based on the plane wave impulse approximation. In the valence space, the HOMO (24a), NHOMO (23a) and orbitals 22a and 18a are selected to study the conformational processes in L-alanine.

1 Introduction

The behaviour of proteins in biological systems at the molecular level is an important area of research, since many biological phenomena can be traced to fundamental properties of the molecular constituents. The native structures of proteins are largely governed by the balance of interactions among the different amino acid residues. Understanding the biological specificity of these species therefore requires insight into the dynamics of amino acids ($\text{NH}_2\text{-CH(R)-COOH}$). The functionality of amino acids varies depending on their conformation [1], which is important in protein stability and three dimensional folding [2]. Therefore, analysis of structural and chemical properties of amino acids under isolated conditions is of great importance.

Alanine ($\text{R}=\text{CH}_3$) is an important prototype for all chiral amino acids, due to its relatively simple structure. In the body, it has a major role in transferring nitrogen from tissue sites to the liver [3]. The conformational behaviour of L-alanine, like glycine [4], has brought many challenges to experimental and theoretical investigations. Up to 13 conformations were predicted from sophisticated *ab initio* calculations [5, 6], several of which were observed experimentally using the millimetre wave (MMW) spectrum of alanine [7] and gas-phase electron diffraction (GED) techniques [8]. The relative stability of these conformers is dependant upon the interplay of various intramolecular hydrogen bonds and electron correlation, as demonstrated for small aliphatic amino acids [4, 6, 9].

In position space, the conformational processes of L-alanine has been investigated, using both DFT and *ab initio* methods [6]. Molecular properties, such as geometries, relative energies and dipole moments have all been effectively predicted using similar

models [5, 6]. The two lowest energy conformations for L-alanine, **I** and **II**, are presented in Fig. 1. Details however regarding the redistribution of electron density and orbital distortion, as a result of these processes, are lacking. This study therefore aims to provide an orbital-based insight into the bonding environment within L-alanine. We focus on information that differentiates the conformers, such as dipole moments and orbital momentum distributions (MDs), using dual space analysis (DSA) [10]. Conformational variations will be discussed in relation to rotations about the $C_{(3)}-O$, $C_{(3)}-C_{\alpha}$ and $C_{\alpha}-N$ bonds, in order to provide an orbital dependant representation of the bonding mechanisms within the two conformations.

2 Methods and Computational Details

Density functional theory (DFT) B3LYP methodology, together with the triple zeta valence polarized (TZVP) basis set [11] was employed for geometry optimization calculations. Electronic structural calculations undertaken in this study used the computational chemistry package of GAUSSIAN03 [12].

Molecular wave functions obtained from single point calculations (B3LYP/TZVP//B3LYP/TZVP) were Fourier transformed into \mathbf{k} -space as momentum distributions (MDs, σ). Under the Born-Oppenheimer approximation, independent particle approximation and the plane wave impulse approximation (PWIA) [13] gives,

$$\sigma \propto \int d\Omega |\psi_j(\mathbf{k})|^2 \quad (1)$$

The integral $d\Omega$ performs the spherical averaging over the initial rotational states, while σ is proportional to the squared momentum space one-electron Dyson orbitals $\psi_j(\mathbf{k})$. The azimuthal angle ϕ has a defined relationship with momentum [13].

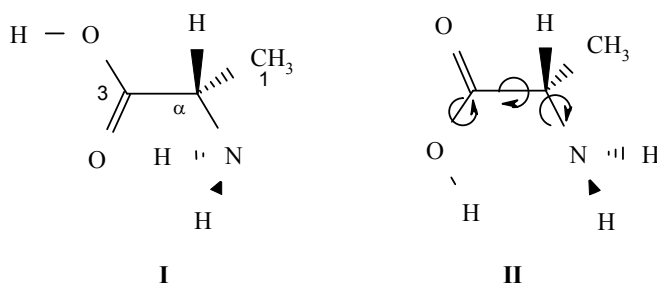


Fig. 1. Chemical structures and numbering system of the L-alanine conformers calculated using the DFT-B3LYP/TZVP model

3 Results and Discussion

3.1 Geometric Correlation of the L-Alanine Conformers

Conformer **I** is the most abundant conformation [1, 7] and represents the global minimum structure on the torsional potential surface. Conformer **II**, the second most

populated conformation (~8:1 ratio in favour of **I**) is produced by three rotations of the $C_{(3)}-C_{\alpha}$, $C_{(3)}-O$ and $C_{\alpha}-N$ bonds in conformer **I**. Selected geometric parameters for these conformations based on the B3LYP/TZVP model are given in Table 1.

The predicted properties for both conformations agree well with previous experimental and theoretical results. All bond lengths lie within 0.02 Å of the experiment, whilst most angular deviations were less than 3°. The exception however is in the description of the $\angle HOC$ angle in **I**. The B3LYP/TZVP model predicted this angle to be 107°, which is 5.3° smaller than the experimental value of 112.3°. A similar deviation was observed by Császár [6] (decrease of 6.1°) when using the MP2/6-311++G** method.

Table 1. Optimized geometrical parameters for conformers **I** and **II** of L-alanine obtained using the B3LYP/TZVP model. Experimental and previous theoretical calculations are provided for comparison.

Parameters	Expt ^a		B3LYP /TZVP		MP2 /6-311++G** ^b	
	I	II	I	II	I	II
$C_{(3)}-C_{\alpha}/\text{Å}$	1.527	1.538	1.528	1.542	1.521	1.533
$C_{\alpha}-C_{(1)}/\text{Å}$	1.536	1.532	1.537	1.532	1.530	1.527
$C_{(3)}-O/\text{Å}$	1.341	1.327	1.360	1.344	1.356	1.342
$C_{\alpha}-N/\text{Å}$	1.453	1.469	1.457	1.477	1.452	1.468
$O-H/\text{Å}$	0.977	—	0.972	0.987	0.968	0.980
$O=C_{(3)}-C_{\alpha}/^{\circ}$	125.6	122.9	125.6	122.8	125.4	122.6
$C_{(3)}-C_{\alpha}-N/^{\circ}$	112.9	108.6	113.5	109.5	113.7	109.4
$H-O-C_{(3)}/^{\circ}$	112.3	—	107.0	104.9	106.2	104.0
$O=C_{(3)}-C_{\alpha}-C_{(1)}/^{\circ}$	—	—	-75.5	-101.8	-76.6	-101.7
$O=C_{(3)}-C_{\alpha}-N/^{\circ}$	-16.6	—	-19.7	-168.4	-20.5	-164.4
$H-O-C_{(3)}-C_{\alpha}/^{\circ}$	180.0	—	178.2	-2.7	176.9	-5.4
Dipole Moment ^c						
μ_x/D	0.62	4.92	0.71	-5.43	0.64	5.24
μ_y/D	1.60	1.40	0.99	-1.38	1.19	1.44
μ_z/D	0.34	0.28	-0.46	-0.65	0.42	0.40
μ/D	1.80	5.13	1.31	5.64	1.41	5.45
$\Delta E/\text{kcal}\cdot\text{mol}^{-1}$			0.0	-0.014	0.0	0.14 (0.03) ^d
$\Delta E_{ZPE}/\text{kcal}\cdot\text{mol}^{-1}$			0.0	0.20	0.0	0.39 ^e

^aElectron diffraction, Ref [8].

^bgeometry, see Ref [6]; Dipole moment see Ref [1].

^cExperimental data based on Millimetre-Wave Spectroscopy, Ref [9].

^dB3LYP/6-311++G**, Ref [6].

^eB3LYP/aug-cc-pVDZ, Ref [14]

In table 1, rotations around the single bonds do not exhibit impact on bond lengths; small angle variations are however observed and apparent changes in dihedral angles exist to accommodate the rotations. This is highlighted by the elongation of the C_{α} -N bond (0.02\AA) and the reduction in the $\angle C_{(3)}C_{\alpha}N$ angle (4°) in **II**, resulting from the weak $\text{OH}\cdots\text{NH}_2$ interaction ($\text{H}\cdots\text{N}$ distance is 1.899\AA). The results obtained in this study agree well with both experimental and previous theoretical calculations.

3.2 Conformational Impact on Stability and Molecular Properties

A number of important molecular properties using DFT [6, 14] and *ab initio* calculations [1, 6] are provided in Table 1 for comparison. With the inclusion of zero-point energy (ZPE), the present method predicts conformer **II** to lie within $0.20\text{ kcal}\cdot\text{mol}^{-1}$ of **I**. This is in satisfactory agreement with the values of 0.14 and $0.39\text{ kcal}\cdot\text{mol}^{-1}$ given by the MP2/6-311++G** and B3LYP/aug-cc-pVDZ [14] (ZPE included) models, respectively.

Properties such as dipole moments are also included in this table as biomolecular conformers that differ by small energies can exhibit profound differences in their anisotropic properties [4], affecting their chemical reactivity. The $C_{(3)}$ - C_{α} , C_{α} -N and $C_{(3)}$ -O bond rotations in **I** cause the total dipole moment to increase significantly from 1.31 D to 5.64 D , which agrees well with the experimental values of 1.80 D (**I**) and 5.13 D (**II**) [9] and the values of 1.41 D and 5.45 D generated using the MP2/6-311++G** model [1]. Such changes indicate that significant charge redistribution is associated with these rotations. This is also evident as μ_x and μ_y have altered considerably as a result of these rotations. For example, μ_x of **I** is 0.72 D and becomes -5.43 D in conformer **II**, whilst μ_z varies significantly without any changes in sign. This trend was observed in glycine [4].

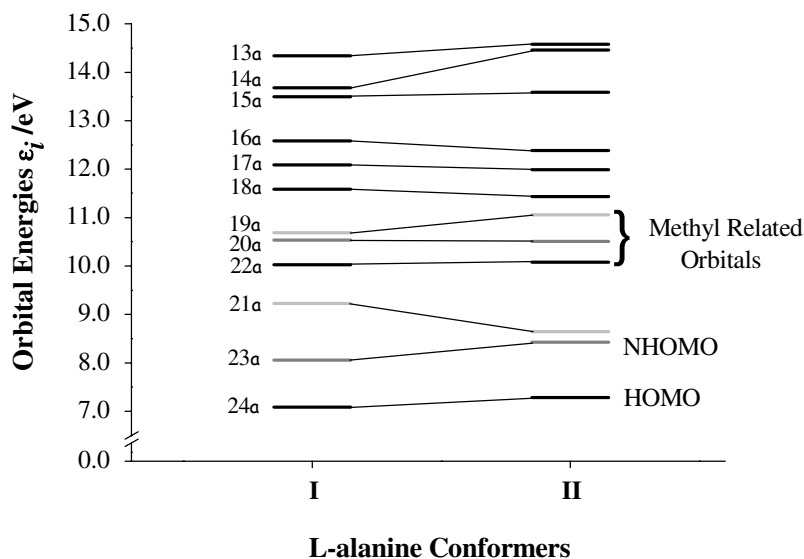


Fig. 2. Outer valence orbital energies for the L-alanine conformer calculated using the B3LYP/TZVP model

3.3 Molecular Orbital Information in Coordinate Space

In their ground electronic states, conformers **I** and **II** (X1A) of L-alanine are both closed shells with singlet states, which have 24 doubly occupied molecular orbitals (MOs). The associated orbital energies for these conformations are given in Fig. 2. From this figure it is clear the three combined bond rotations in **I** cause major energy variations in orbitals 14a, 19a, 22a and 23a, whilst visible changes to orbitals 13a, 16a, 18a and 24a are observed. From information provided thus far it is still unclear as to the changes in the nature of bonding from these rotations. This will be explored in more detail in the next sections using information from momentum space.

3.4 Orbital MDs of L-Alanine in Valence Space

Orbital momentum distributions (MDs) of L-alanine in the outer valence space have been calculated in the present study. Both conformations belong to the C_1 point group symmetry and their Dyson orbitals are therefore correlated. Four representative orbitals were selected to demonstrate interesting structural information of L-alanine (Fig. 3). Orbitals of other conformers which exhibit the most significant variation with respect to the global minimum structure of L-alanine are considered to be the fingerprints of rotation around a particular bond.

In Figure 3(a), orbital MDs related to MO 24a (HOMO) indicate significant mixed 's-like' and 'p-like' characteristics in both conformations. The associated charge distributions of the Dyson orbital reveal that in **I** these contributions are predominately from the lone-pair on the N atom and N-H bonds of the amino group. The rotations enhance contributions on the lone-pair of the O in $O=C_{(3)}$, whilst reducing the charge density on the amino group. The latter reduces the orbital density of H atoms in the $-NH_2$ group, causing the momentum distributions to drop in the low momentum region. This may have significant implications in the chemical reactivity of these conformers as any structural manipulation, i.e., bond breaking or bond formation occurs within these frontier orbitals.

In MO 23a (NHOMO), the nature of bonding in the two conformations are essentially reversed. That is, the p-like contributions in **I** are primarily from the lone pair on O of $O=C_{(3)}$, whereas in conformer **II** the contributions stem from the amino group [Fig. 3(b)]. Sizeable s-contributions are also present along the C_α -N bond, contributing to the enhancement of the orbital MDs in the region of low momentum. This demonstrates that orbital MDs can serve as a quantitative property in the identification of electron redistribution. Similar trends to the HOMOs and NHOMOs were observed in the glycine conformers [4], where rotation of the $C_{(3)}$ - C_α and C_α -N bonds were shown to cause the most perturbation to the orbitals in this region.

In contrast, MO 22a in Fig. 3(c) demonstrates that the electron charge distributions concentrating in the HO-C=O region in **I** did not change with the single bond rotations, even with the rotation of the HO-C bond. A small enhancement of charge density on the NH_2 group in **II** causes the small shift of orbital MDs towards the low momentum region, as indicated in Fig 3(c). In both conformations, this MO is clearly a π -bond with the nodal plane being the molecular plane, which is formed by the non-hydrogen atoms (excluding the methyl group). These orbitals are related to the anti-symmetric orbitals of a'' symmetry found in glycine [4, 15].

In comparison to MOs 24a and 23a (Fig 3 (a) and (b)), where the discrepancies in the momentum distributions occur at zero momentum due to the hydrogen 1s AO contributions, orbital 18a in **I** and **II** differ in the region of medium momentum at approximately $\phi = 10^\circ$ (Fig. 3(d)). This orbital correlates to orbital 14a' in glycine [4], which was shown to be the fingerprint orbital for all single bond rotations. In MO 18a, rotation of $C_{(3)}-O$ in **I** caused the strong delocalised coverage along the $H-O-C_{(3)}-C_\alpha$ framework to break into more isolated groups of $H-O-C_{(3)}$, $C_{(3)}=O$ and $C_{(3)}-C_\alpha-C_{(1)}$. These changes are reflected in the associated orbital MDs in Fig. 3(d).

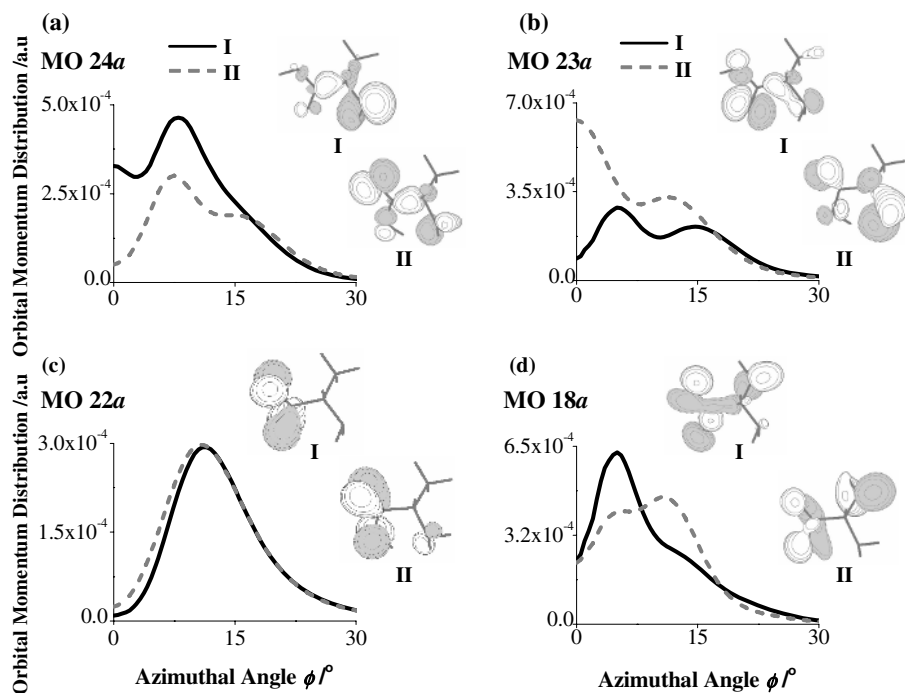


Fig. 3. Momentum distributions and charge distributions of selected orbitals (24a, 23a, 22a and 18a) for conformers **I** and **II**. Dyson orbitals are plotted using Molden [16].

4 Conclusion

Rotational processes of the two lowest energy conformers of L-alanine (XA) have been studied using dual space analysis (DSA). Changes to the geometric parameters after the $C_{(3)}-C_\alpha$, $C_\alpha-N$ and $C_{(3)}-O$ bond rotations in the global minimum (**I**) largely resulted from the related dihedral angles reflecting the rotations. Weak interactions, such as $O-H \cdots NH_2$ also contribute to the changes in structure. Between the two conformations, redistribution of the electron charge density was also found to vary significantly in certain orbitals.

Variations in both binding energy spectra and orbital momentum distributions in L-alanine were observed as a result of the rotations. Orbitals exhibiting significant changes in binding energy however did not necessarily vary significantly in the shape of their electron density (orbital MDs) and vice versa. While the HOMO (24a) and orbital 18a did not exhibit major changes in their binding energies with respect to their conformations, however these orbitals experienced significant variations in their electron density, as reflected by their orbital MDs. Orbital 22a receives significant changes in binding energy but its electron density does not vary much with respect to conformation. Using dual space analysis, this investigation therefore provides a unique insight into the structural effects of alanine upon rotation.

Acknowledgements. The authors thank the Vice-Chancellor's Strategic Research Initiative Grant of Swinburne University of Technology and the Australian Partnership for Advanced Computing for the use of the National Supercomputing Facilities.

References

1. Blanco, S *et al.*, *J. Am. Chem. Soc.*, **126**, (2004), 11675.
2. Baldwin, T.; Lapointe, M in *The Chemistry of Amino Acids. The Biology Project [Online]*, http://www.biology.arizona.edu/biochemistry/problem_sets/aa/aa.html, 2003.
3. Rennie, M. J. in *Physical Exertion, Amino Acid and Protein Metabolism and Protein Requirements; Proteins and Amino Acids*. Washington, DC: National Academy Press, (1999).
4. Falzon, C. T., Wang, F. *J. Chem. Phys.*, **123**, (2005), 214307.
5. (a) Cao, M., *et al.*, *J. Mol. Struct: THEOCHEM*, **332**, (1995), 251. (b) Godfrey, P. D., *et al.*, *J. Mol. Struct.*, **376**, (1996), 65. (c) Kaschner, R., Hohl, D. *J. Phys. Chem. A.*, **102**, (1998), 5111.
6. Császár, A. G. *J. Phys. Chem.*, **100**, (1996), 3541.
7. Godfrey, P. D., *et al.*, *J. Am. Chem. Soc.*, **115**, (1993), 9687.
8. (a) Iijima, K., Beagley, B. *J. Mol. Struct.*, **248**, (1991), 133. (b) Iijima, K., Nakano, M. *J. Mol. Struct.*, **485-486**, (1999), 255.
9. Lesarri, A. *et al.*, *Angew. Chem. Int. Ed.*, **43**, (2004), 605.
10. Wang, F. *J. Phys. Chem. A.*, **107**, (2003), 10199.
11. Godbout, N., *et al.*, *Can. J. Chem.*, **70**, (1992), 560.
12. Frisch, M. J., *et al.*, *Gaussian Inc*; Wallingford: CT, (2004).
13. McCarthy, I. E., Weigold, E. *Rep. Prog. Phys.* **54**, (1991), 789.
14. Stepanian, S. G., *et al.*, *J. Phys. Chem. A.*, **107**, (1998), 4623.
15. Falzon, C. T., Wang, F., Pang, W. N., *J. Phys. Chem. B.*, **000**, (2006), 0000.
16. Schaftenaar, G., Noordik, J. *J. Comput-Aided Mol. Design.*, **14**, (2000), 123.

***Ab initio* Modeling of Optical Properties of Organic Molecules and Molecular Complexes**

Vladimir I. Gavrilenko

Center for Materials Research, Norfolk State University, 700 Park Avenue, Norfolk VA
23504, USA
vgavrilenko@nsu.edu

Abstract. Electronic excitations are key points of most of the commonly measured optical spectra. The first principle studies of excited states however require much larger effort than computations of the ground state reliably reproduced by the density functional theory (DFT). In present work computation of optical functions of organic molecular complexes is studied. The system of independent particles excited by external light field is considered within perturbation theory (the random phase approximation, RPA). Optical response functions are calculated using *ab initio* pseudopotentials theory. Results of predicted optical absorption associated with organic semi-conducting conjugated polymers, poly-phenylene-vinylenes (PPV), are presented. Effects of different corrections to the DFT improving accuracy are considered. Results are discussed in comparison with available experimental data.

1 Introduction

It has been demonstrated for decades that Kohn-Sham density functional theory (DFT) realistically predicts electronic structure of different systems, such as atoms, molecules, and solids [1 to 3]. Various generalized gradient approximation (GGA) methods describing exchange and correlation (XC) interaction have been shown to systematically improve the local density approximation (LDA) predictions of equilibrium atomic geometries [3]. In atomic systems (like large single molecules, inorganic solids etc.) the geometry optimization study primarily requires accurate prediction of the ground state which is well reproduced within a standard DFT approach. Electronic excitations are key points of most of the commonly measured optical spectra of organic materials (see [4] and references therein). The first principle studies of excited states however require much larger effort than computations of the ground state. The DFT eigen energies underestimate gaps between bonding and anti-bonding states which requires quasi-particle (QP) correction [2,3]. It has been proved that in inorganic solids [5] and organic polymers [2] the QP correction results in substantial overestimates of the gaps. The situation is substantially improved by additional inclusion of excitonic (many-body) interaction into Hamiltonian of the system [2,3]. This state-of-the-art DFT+QP many-body theory predicts excitation energies in organic polymers with a good accuracy [2]. Such approach however

requires large scale computations and it is very time consuming. On the other hand the corrections for QP and exciton interaction in electron eigen energies could be included through the scissors operator [3,5].

In this work a first principle method to predict optical functions of complex organic molecules and molecular complexes based on *ab initio* pseudopotential (PP) theory is described. Corrections for QP shift and excitonic effects are incorporated through the scissor operator. This method could be used to study electronic structure and optical functions of complex molecules and complexes which is demonstrated for organic polymer calculations.

2 Method

Optical absorption spectra of organic conjugated poly-phenylene-vinylenes (PPV) polymers with linear decacyl ($C_{10}H_{21}$) groups (donor RO-PPV block) are calculated using *ab initio* pseudopotentials method. For this system an infinitely long PPV polymer chain is considered. This approach is distinctly different from that traditionally used in quantum-mechanical calculations [4]. Advantages of this approach were demonstrated for trans-polyacetylene and simple PPV polymers [2]. Recently we applied this method for optical absorption study of Rhodamine 6G (R6G) dye molecular aggregates [6].

Here both commercial [7] and research [8] computational packages based on *ab initio* PP are used. In order to apply pseudopotentials in molecular system we use the super-cell method [9]. The electron energy structure and eigen-functions are calculated within DFT-GGA using fully separable *ab initio* PP generated according to the Troullier-Martin scheme [10]. Energy cut-off up to 60 Ry is taken to generate fully separable PP [8]. Convergence of the ground state has been proved by test calculations with energy cutoff (E_{cut}) up to 100 Ry for R6G single molecule.

Accuracy of better than 10% for eigen values is achieved with $E_{cut} = 45$ Ry. This value is used in this work in order to avoid complexity. For atomic relaxation of single molecule and molecular dimers the super cell method is used. The Perdew-Burke-Ernzerhof (PBE) functional was used to model XC interaction [11].

Optical absorption spectra are calculated within RPA approach [5, 8, 9] according to $\alpha(\omega) = 2\omega k(\omega)$ [12]. The extinction coefficient k is determined through real and imaginary parts of the dielectric permittivity function $\epsilon(\omega) = \epsilon_1(\omega) + i\epsilon_2(\omega)$. The evaluation of $\epsilon_{\alpha\beta}(\omega)$, ($\alpha, \beta = x, y, z$), in this work is based on the independent particle approximation. We consider the effect of gauge invariance for computation of $\epsilon_{\alpha\beta}(\omega)$ (details are given in Appendix). The time dependent linear optical susceptibility function follows from the equation of motion for the density matrix (the RPA picture) [5,9]. The imaginary part of the function of $\epsilon_{\alpha\beta}(\omega)$ in Coulomb (or velocity) gauge and in length gauge is given by Eq (16) or Eq (17) respectively. The real part is calculated using Kramers-Kronig formula [12].

3 Results and Discussion

In order to study the effect of the gauge invariance on predicted optical spectra of organic molecules the optical absorption spectra of R6G molecular dimmers [6] and of RO-PPV polymer are calculated using both Coulomb (velocity) and length gauge (Eqs (16) and (17)). We found [6] that length gauge for R6G calculations is preferable. The level of 10% accuracy of the convergence of the optical functions could be reached faster than with velocity gauge. This agrees with Rautian's [4] finding. He showed that length gauge is more convenient for confined system, where if using Coulomb gauge, one still needs to sum over wide spectral range. On the other hand we found that for delocalized system (RO-PPV polymer) velocity gauge is more convenient. This approach allows easy corrections to the optical response function. In particular, for several cases in materials optics one needs to incorporate non-negligible contributions to optical response functions caused by intra-band electron transitions (e.g. for molecular and/or atom surface adsorption, surface/interface contributions etc. [9]) Straightforward use of length gauge in this case causes divergences, which require special treatment [3,9], consequently the Coulomb gauge is preferable.

The Coulomb gauge is very useful by evaluation of Hamiltonian matrix elements in a plane wave basis by using *ab initio* pseudopotential method [8]. In this case optical response functions (both linear and nonlinear) are evaluated in particularly simple way, which is very important for complex systems [5,9]. However strictly speaking in the optics of materials one can not use momentum operator for the free particle (6). In many-particle system (like molecules or solids) electron experience not only on-site (local) excitations, but also effect of the other particles, which is essentially non-local. This results in the non-locality of momentum operator. Nature of this effect could be understood from the following consideration.

Quite generally the many particles unperturbed Hamiltonian is given by:

$$H = \frac{\hat{p}_0^2}{2m} + V_{loc}(\vec{r}) + \sum(\vec{r}, \vec{r}') \quad (1)$$

where operator $\sum(\vec{r}, \vec{r}')$ represents non-local part of the potential energy of the Hamiltonian.

Momentum (or velocity) operator is given by:

$$\hat{p} = m \frac{d}{dt} \vec{r} = m \frac{i}{\hbar} [\hat{H}, \vec{r}] = m \frac{i}{\hbar} \left[(\hat{H}\vec{r}) - (\vec{r}\hat{H}) \right] \quad (2)$$

Taken into account that $[V_{loc}(\vec{r}), \vec{r}] = 0$ and using following relation for the kinetic part of the Hamiltonian:

$$(\vec{\nabla}^2 r - r \vec{\nabla}^2) \psi = \vec{\nabla}(\psi + r \vec{\nabla} \psi) - r \vec{\nabla}^2 \psi = 2 \vec{\nabla} \psi \quad (3)$$

The equation (2) results in:

$$\hat{p} = -i\hbar\vec{\nabla} + m\frac{i}{\hbar}[\sum(\vec{r}, \vec{r}'), \vec{r}] \quad (4)$$

Last term in (4) represents correction to the momentum operator of the free particle (6) for non-locality of the potential energy (for the particle in molecule or solid). This contribution results in redistribution of the oscillator strengths which substantially improves predicted shape of optical spectra by bringing them closer to experimental data, as it has been demonstrated for inorganic group IV materials in [13]. This should be borne in mind by analysis of the predicted spectral shape.

Equilibrium atomic configuration of organic semi-conducting conjugated polymer RO-PPV ($R=C_{10}H_{21}$) is determined by geometry optimization through minimization of the total energy and it is shown in Fig. 1.

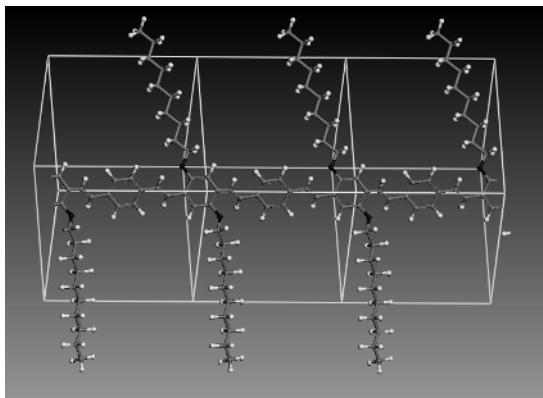


Fig. 1. Equilibrium atomic geometry of RO-PPV conjugated polymer determined through total energy minimization method. By the white line super cell units ordered in one dimensional chain are shown.

Optical absorption spectra of RO-PPV conjugated polymers are calculated using Coulomb gauge from Eq. (16). The self-consistently calculated eigen-functions and eigen-energies for equilibrium geometries are used as inputs. Calculated optical absorption spectrum of PPV single chain conjugated polymer is shown in Fig. 2. Absolute values of the absorption coefficient are normalized according to the molecular concentration followed from the dimension of the super cell.

Optical absorption of the light polarized along polymer chain $\vec{E} \parallel \vec{z}$ (see Fig. 2) is more than an order of magnitude stronger than that for perpendicular geometry. Absorption peak around 440 nm arises from the excitations of delocalized π -electrons from benzene rings. The basic PPV polymer chain does not show any optical absorption for perpendicular geometry in a visible range [2]. The RO groups slightly change symmetry of the delocalized π -electrons thus allowing non-zero optical absorption for $\vec{E} \perp \vec{z}$ in a visible range. The quasi-particle correction is applied in the

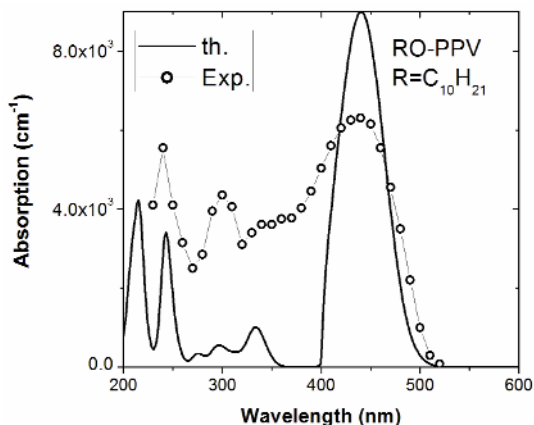


Fig. 2. Calculated optical absorption spectrum of RO-PPV conjugated polymer calculated for parallel ($\vec{E} \parallel \vec{z}$) geometry. Symbols represent measured optical absorption spectra [14].

form of scissor operator: the predicted anti-bonding electronic states are increased in energy by 0.79 eV to match the 440 nm measured absorption peak. With this *ad hoc* QP correction the predicted optical absorption spectrum of RO-PPV correctly represents all main features seen in the measured spectrum (see Fig. 2). Calculated absorption spectrum is close to that observed in similar PPV polymers [4, 15]. However absolute intensities of the absorption peaks deviate strongly from experimental data. Similar situation was reported on simple polymers by using first principle many-body excitonic theory [2]. Better agreement we obtained for R6G dye molecular dimers [6]. The reason for the discrepancy between the shapes of predicted and measured absorption spectra could be the neglect of the momentum operator nonlocality as discussed above. Inclusion of this effect into optical calculation improves agreement with experimental spectral shape for inorganic solids [13].

In [2] full *ab initio* computations of excitation energies of some simple polymers demonstrated importance of excitonic effects in optics. Such large scale computations within first principle theory (the DFT with QP and many-body exciton corrections [2]) are still challenging for computational physics of complex organic materials. Therefore the first principle pseudopotential method with the *ad hoc* QP scissor correction for DFT-GGA excitation energies in organic polymers is a reasonable compromise between accuracy and complexity of computations, as demonstrated in this work. Results of the present work as well as that presented in the literature show that overall agreement between predicted and experimental optical spectra in polymers is worse than in solids. One important reason is the reduction of the translational symmetry to only one dimension in polymers. Consequently the increase of the unit cell requires more extensive computational work to achieve the same accuracy level as in three dimensions. Further development of computational methods combining that of physics and chemistry may solve the problem.

4 Conclusions

Method of optical functions calculations of organic molecular complexes based on *ab initio* pseudopotentials is described. Optical excitation energies are correctly predicted using DFT-GGA method with scissor operator for quasi particle correction. Numerical results obtained for RO-PPV conjugated polymer are in reasonable agreement with experimental data. Effects of gauge invariance and non-locality of optical momentum operator on predicted optical response functions are analyzed.

Acknowledgement

Author is thankful for discussions with M. Noginov and C. Bonner. This work is supported by NSF grant CREST supplement No. 0520208.

References

1. Kohn W., Sham L., J. Phys. Rev., 140 (1965) A1133-A1143
2. Rohlfsing M., Luie S. G., Phys. Rev. Lett. 82 (1999) 1959-1962
3. Onida G., Reining L., Rubio A., Rev. Mod. Phys. 74 (2002) 601-633.
4. Barford W., Electronic and Optical Properties of Conjugated Polymers, Oxford University Press, New York (2005)
5. Gavrilenko V. I., Bechstedt F., Phys. Rev. B 55 (1997) 4343-4352
6. Gavrilenko V. I., Noginov M. A., J. Chem. Phys. 124 (2006) 44301-44306
7. Material Studio Modeling 3.2, Accelrys Software Inc. 2004.
8. Fuchs M., Scheffler M., Comput. Phys. Commun., 16 (1999) 1-18
9. Gavrilenko V. I., Phys. Status Solidi (a), 188, (2001) 1267-1280
10. Troullier N., Martins J. L., Phys. Rev. B, 43 (1991) 1993-1998
11. Perdew J. P., Burke K., Ernzerhof M., Phys. Re. Lett., 77 (1996) 3865-3875
12. Yu P., Cardona M., Fundamentals of Semiconductors, Springer-Verlag, Berlin Heidelberg New York (2001)
13. Adolph B., Gavrilenko V. I., Tenelsen K., Bechstedt F., Del Sole R., Phys. Rev. B, 53 (1996) 9797-9808.
14. Bonner C. E., Jr., Charter S., Lorts A., Gavrilenko V.I. SPIE Int. Conf. on Optics and Photonics, (San Diego 2006), (submitted).
15. Frolov S., Bao Z., Wohlgenannt M., Vardeny Z.V., Phys. Rev. B 65 (2002) 205209-205218

Appendix: Effect of Gauge Invariance on Evaluation of Optical Susceptibility Functions

Perturbed Hamiltonian of the particle in electro-magnetic field of light is given by:

$$H = \frac{1}{2m} \left[\hat{p} + \frac{e}{c} \vec{A}(\vec{r}, t) \right]^2 + V(\vec{r}), \quad (5)$$

The momentum operator of free particle is given by:

$$\vec{p}_0 = -i\hbar\vec{\nabla} \quad (6)$$

Using a trial function we have:

$$\left(\vec{A}\hat{p}_0\right) - \left(\hat{p}_0\vec{A}\right) = i\hbar \operatorname{div}\vec{A} \quad (7)$$

Assuming Coulomb gauge ($\operatorname{div}\vec{A} = 0$) and bearing in mind that in the light wavelength range up to vacuum ultraviolet the quadratic term on vector-potential in (5) is very small [12], we have:

$$H = \frac{\hat{p}_0^2}{2m} + \frac{e}{mc} \left(\vec{A}\hat{p}_0\right) + V(\vec{r}) = H_0 + H_{\text{int}} \quad (8)$$

with

$$H_{\text{int}} = \frac{e}{c} \left(\vec{A}\hat{p}_0\right) = \frac{e}{mc} A_0 e^{i(\alpha x + \vec{q}\vec{r})} (\vec{\tau}\hat{p}_0) = -i \frac{e}{m\omega} E_0 e^{i(\alpha x + \vec{q}\vec{r})} (\vec{\tau}\hat{p}_0) \quad (9)$$

where τ is a unity vector, and for the light wave we have:

$$\vec{E} = -\frac{1}{c} \frac{\partial \vec{A}}{\partial t} = i \frac{\omega}{c} \vec{A} \quad (10)$$

Let us consider now the full momentum matrix elements of p , determined on the eigen functions of the Hamiltonian (5):

$$H\psi_n = E_n\psi_n \quad (11)$$

We have:

$$\begin{aligned} p_{\text{in}} &= m \left\langle l \left| \frac{d\vec{r}}{dt} \right| n \right\rangle = m \frac{i}{\hbar} \left(\sum_{n'} H_{\text{in}'} r_{n'l} - \sum_{n''} r_{\text{in}''} H_{n''n} \right) \\ &= m \frac{i}{\hbar} (E_l - E_n) r_{\text{in}} = im\omega_{\text{in}} r_{\text{in}} \end{aligned} \quad (12)$$

This expression is general. In the length gauge the interaction Hamiltonian is given by [9]:

$$H_{\text{int}} = e\vec{r}\vec{E} = eE_0 e^{i(\alpha x + \vec{q}\vec{r})} (\vec{\tau}\vec{r}) \quad (13)$$

Based on (9) and (12) the ratio between Coulomb and length gauge interaction Hamiltonians is given by:

$$\frac{H_{\text{int}}^C}{H_{\text{int}}^L} = -i \frac{1}{m\omega} \frac{(\vec{\tau}\hat{p})}{(\vec{\tau}\vec{r})} \quad (14)$$

$$\frac{\langle l | H_{\text{int}}^C | n \rangle}{\langle l | H_{\text{int}}^L | n \rangle} = -i \frac{1}{m\omega} \frac{im\omega_{\text{ln}} \vec{r}_{\text{ln}}}{\vec{r}_{\text{ln}}} = \frac{\omega_{\text{ln}}}{\omega} \quad (15)$$

Last expression agrees with that given in [4]. It shows that in resonance (if $\omega_n = \omega$) both Coulomb and length invariance are equivalent, but out of resonance they are different. Based on time dependent perturbation theory expressions for imaginary part of dielectric permittivity in both gauges are given by [9, 12] for Coulomb and length gauges, respectively:

$$\varepsilon_2(\omega) = \left(\frac{2\pi e}{m\omega} \right)^2 \sum_{l,n} |p_{\text{ln}}|^2 \delta(E_l - E_n - \hbar\omega) \quad (16)$$

$$\varepsilon_2(\omega) = (2\pi e)^2 \sum_{l,n} |r_{\text{ln}}|^2 \delta(E_l - E_n - \hbar\omega) \quad (17)$$

A Framework for Execution of Computational Chemistry Codes in Grid Environments

André Severo Pereira Gomes¹, Andre Merzky², and Lucas Visscher¹

¹ Department of Theoretical Chemistry, Faculty of Exact Sciences Vrije Universiteit Amsterdam, De Boelelaan 1083, 1081 HV Amsterdam The Netherlands

² Department of Computer Science, Faculty of Exact Sciences Vrije Universiteit Amsterdam, De Boelelaan 1081A, 1081 HV Amsterdam The Netherlands
a.gomes@few.vu.nl, merzky@cs.vu.nl, visscher@chem.vu.nl

Abstract. Grid computing is a promising technology for computational chemistry, due to the large volume of calculations involved in applications such as molecular modeling, thermochemistry and other types of systematic studies. Difficulties in using computational chemistry codes in grid environments arise, however, from the fact that the application software is complex, requiring substantial effort to be installed on different platforms. Moreover, these codes depend upon task-dependent sets of data files to be present at the execution nodes. Aiming to improve the usability of different quantum chemistry codes in the distributed, heterogeneous environments found in computational grids, we describe a framework capable of handling the execution of different codes on different platforms. This framework can be divided into three independent parts, one dealing with the mapping of a calculation to a set of codes and the construction of execution environments, one dealing with the management of grid resources, and one that takes care of the heterogeneity of the environment. The suitability of this framework to tackle typical quantum chemistry calculations is discussed and illustrated by a model application.

Keywords: Grid Computing, Computational Chemistry, Grid(lab) Application Toolkit, Heterogeneous Environment, Command-line Interface.

1 Introduction

Grid computing has evolved rapidly over the past few years, and has reached a stage where scientists outside the field of computer science are beginning to explore the potential benefits of these new technologies. Computational chemistry is an application area where grid computing may have a significant impact, due to its ever-increasing demand for computational resources. While grid computing may not be suitable for all problems taken up by computational chemists, due to memory or other hardware requirements, it will likely be very useful in increasing throughput in cases where calculations are not truly expensive in themselves but are to be performed in large numbers. Examples of these are potential

energy surface (PES) scans, accurate thermochemical calculations, combinatorial design of new compounds, or in finite temperature statistical averaging of molecular properties.

The relatively few initiatives to use grid computing in computational chemistry have so far involved either the modification of an existing code to use some of the functionality of a grid middleware[1, 2], such as Globus[3], or the creation of portals[4, 5], user interfaces and/or specialized middleware to handle both grid-related tasks, such as authentication, as well as application-specific tasks, such as the creation of inputs and visualization of results[6, 7, 8]. These efforts, typically restricted to one or a few codes, have a significant drawback because they rely on a specific middleware to access grid resources (some projects use the Globus toolkit while other use UNICORE[9]), demanding major revision as the middleware evolves. It is not clear, moreover, how easy it would be in these approaches to support different codes, particularly in heterogeneous environments. It is also not clear at this stage how to achieve high throughput in environments that rely solely on graphical interfaces to prepare and/or submit jobs.

Having these issues in mind, in this paper we describe a framework that allows the use of grid resources by computational chemists, which: (a) already has support for different codes and is easily extendable to include new application software or support more complex work flows ; (b) allows for use of heterogeneous grids; (c) has no dependence on a specific middleware; and (d) has a simple user interface. A standard command-line tool is supplied, that can be replaced by graphical user interfaces or grid portals if desirable.

2 Implementation Details

The framework constructed can be divided in four individual components which are responsible for: (a) handling the use of binary executables and platform-dependent data, in order to allow for the use of grid environments with varying degrees of hardware and/or software heterogeneity; (b) performing tasks related to the grid environment, such as requesting resources, scheduling jobs for execution, cancelling jobs, verifying a job's status, transferring files and so on; and (c) creating jobs (which in this context are characterized by the collection of the input and eventual restart data for a calculation with a program, or collection of programs). These are implemented in the Python programming language, and are integrated via a basic command-line user interface.

2.1 User Interface

The standard interface between the user and the stages of job creation, submission and management is a command-line tool, with which it is possible to create, submit, cancel or query information from jobs running on grid resources. Job creation does not depend on the other actions and needs only little information about the grid environment that will be used. The other actions inherently require more knowledge about the specific grid and middleware deployments.

In an interactive procedure the user is asked to first specify some global preferences for hardware resources (minimal memory requirements, preferred type of architecture, etc.), software resources (computational chemistry code to be used) and standard location of the input files. This step is typically done only once and serves to restrict the amount of data that is to be given in the production stage. The global preferences may later be adjusted by creating a *project* that defines a label for a set of specific preferences for hardware, software and location that is optimal for the type of work that is to be carried out. Also this configuration step is done interactively, and is only necessary when a new project is created or when the user want to adjust its defaults. In a given project one may then easily define many *jobs* by giving only the specific input and workflow for the individual tasks. One may thereby combine job creation and submission if desired.

2.2 Handling Platform-Dependent Data

An important issue one faces when trying to use computational chemistry codes, particularly in heterogeneous grid environments, is how to select an adequate executable. While in some fields it might be possible to generate the correct binary from source code at the execution host, this is impractical for computational chemistry codes, that tend to be fairly large (a common figure is 500k lines or larger of FORTRAN code). In such a case an “on the fly” compilation can easily dominate the total CPU time used. Moreover, most codes are difficult to compile without user intervention, because they usually contain legacy (and sometimes poor quality) code and usually require special mathematical libraries that may not be available at the execution host.

Because of these difficulties, we have opted to handle heterogeneous environments by precompiling different binaries for different combinations of operating systems and machine architectures that are made available at one or more remote locations via `http` or `gridftp`. Upon execution, the job will determine the operating system and architecture of the execution host and retrieve the appropriate binary (in compressed form), using tools like `globus`' file transfer facilities, or other transfer mechanisms, such as `curl` or `scp`, depending upon the way the general or project preferences were set in the configure step(s).

One problem that can not be solved at the moment is the fact that commercial codes often have a licensing scheme that is incompatible with use in computational grids that connect more than one institution. ADF[10], for instance, has a licensing scheme that verifies at runtime the place of execution, and will stop if no valid license file is found. Obtaining a license that enables such codes to be used within the whole grid infrastructure is too expensive in the current setup, while vendors will be reluctant to relax the license checking because it will increase the possibility of unauthorized use. Before grids can be widely employed in production work it is thus necessary that vendors and middleware developers come up with more suitable licensing schemes, e.g. based on a “pay per use” system.

2.3 Use and Management of Grid Resources

Interaction with grid resources, i.e. for the transfer of files, submission of jobs, cancellation of jobs, resource management and so on, is handled by the Grid Application Toolkit (GAT) [11], developed in the GridLab project. This toolkit was chosen since it provides several attractive features, such as: (a) an API that closely resembles UNIX system calls, which makes the function calls to the corresponding middleware's functionality easy to understand and relatively simple to use, (b) the freedom to easily change grid middleware if necessary for a particular case or grid deployment; (c) a built-in persistent database, referred to as the advert service, which keeps track of jobs' status, the location of files, and execution hosts; and (d) the availability of wrappers in Python to the C reference implementation. The use of Python wrappers makes it easier to maintain the code and, more importantly, to later reuse components in other projects within our group, such as graphical user interfaces.

2.4 Job Creation

The job creation module gathers all input file(s) for a given computational code and uses the information contained in these file(s) to generate a script for the execution of this job. In order to do so a two-step procedure for job creation was implemented. In the first step, all data to be transferred (such as execution scripts, input files, special basis sets, and other non-executable data) is collected in a tar-archive. This archive is included in the script during its assembly on the second step. This script, which encapsulates all the remotely executed commands, then acts as an application manager-like "container" for the grid-unaware codes such as those generally used in chemistry.

The basic operations involved in the first step are: (a) scan the input file(s) to determine the data dependencies that should be met (required basis set files, libraries, etc); (b) collect all required data files in a temporary directory; and (c) create a compressed tar file from the directory and convert this to an encoded ASCII format via utilities such as `uuencode`; (d) remove the temporary directory. In the implementation we minimized the amount of data to be archived, so that the data transfer to a remote host should take little time, also on slow networks. In line with this philosophy, we avoid including executable binaries in the container script, so that it is possible to handle large numbers of jobs without unnecessary replication of data on the submission host. We are currently working to incorporate more complex workflows that allow for use different codes at the remote host. In this case, the input is a "control file" specifying inputs for each calculation, and the actions and dependencies that connect the individual calculations.

In the second step a Bourne shell script is constructed automatically, taking into account special demands of the code, or collection of codes, to be used. The encoded archive is inserted into this script and, upon execution at the remote host, decoded, uncompressed and unarchived into a directory private to the job. This "sandboxing" ensure that calculations will not interfere with other jobs that may be executing at the same host. Moreover, it also allows the location of output and intermediate files (restart data etc) to be easily determined on the

remote host, so that GAT's functionality may be used to inspect and copy files to/from the remote host during a calculation.

By using the "container" approach, which is in fact an acknowledged practice in grid environments[12, 13], it is possible in principle to execute any existing code running under UNIX-like environments invoked for calculation as a command-line application (either directly or with the help of shell scripts). As this situation is found in most of the computational chemistry codes available to date, this procedure can handle the execution of any code in platforms for which these are supported. Currently ADF[10], Dalton[14], Dirac[15] and Dacapo[16] are supported, but since code-dependent actions are largely confined to scanning and setting shell environment variables, it is relatively easy to support additional codes within this framework. A second feature that makes this approach attractive is the control over the moment when binaries and other data are fetched, so that these actions could be postponed until they are actually needed, thus allowing for a more balanced communication overhead.

3 Application Tests

The framework introduced here has been largely developed and tested for the currently supported codes on the DAS-2 computer[17], a distributed but architecturally homogeneous Linux machine which is part of the Dutch Grid initiative. However, as one of its strengths lies in handling heterogeneous systems, we also present some results from calculations run on the GridLab testbed[18] and on a small-scale test grid assembled within our Theoretical Chemistry Department (TC-VU) using Macintosh desktop computers as execution hosts and the Xgrid tool as the middleware[19].

Since we are concerned with evaluating the framework, we have simply taken representative examples from the test sets supplied with the quantum chemistry codes and executed them at the platforms mentioned above. Important information that was to be gained in these tests is the effect of communication overhead, due to transferring of files to remote locations, on the total execution time. Table 1 summarizes the timing results obtained for ADF, Dirac and Dalton job performed at different geographical locations. The binary repository is located within the Theoretical Chemistry Department network and, apart from calculations performed at the departmental grid, the DAS-2 head node at the Vrije Universiteit (`fs0.das2.cs.vu.nl`) is used as a submission host.

From the results it is clear that, even for the short jobs considered, the amount of time spent on executing a given computational chemistry code dominates the total execution time for the job. Moreover, since the actions related to setting the execution environment on the remote hosts, such as transferring binary files, appear to take a roughly constant time for a given machine and type of calculation, it is reasonable to assume that already for moderately long calculations the grid-related overhead involved will become negligible.

The role of high-speed interconnection between grid locations is also evident from the results. Very similar results are obtained for the departmental grid

Table 1. Total job wall times (in seconds) and the total calculation/total job time ratios for ADF, Dalton and Dirac test runs in different grid environments. Here *TC VU* denotes the departmental grid; *DAS2 VU* and *DAS2 UU* nodes within DAS-2 (`fs0.das2.cs.vu.nl` and `fs4.das2.phys.uu.nl`); and *Gt PL* nodes within the Gridlab testbed (`eltoro.icis.pcz.pl`).

Code	Test	<i>TC VU</i>		<i>DAS2 VU</i>		<i>DAS2 UU</i>		<i>Gt PL</i>	
		time	ratio	time	ratio	time	ratio	time	ratio
ADF	C2H4 TDCDFT	154	.92	391	.95	388	.94	259	.81
	EPR-SOO	339	.91	337	.92	325	.94	207	.71
	NMR VOCl3	364	.91	597	.95	589	.96	387	.83
	PF3 Nmr	380	.94	824	.97	822	.97	560	.88
	VO Collinear	506	.96	775	.98	782	.98	1264	.96
Dalton	prop vibana	455	.96	568	.99	587	.99	770	.96
	walk solvmag	206	.94	563	.99	568	.99	713	.96
	cc e triplet	189	.92	236	.97	215	.97	195	.88
	dft qr nosym	202	.90	375	.97	362	.98	509	.94
Dirac	b3lyp ch4	251	.99	527	.994	524	.994	778	.99
	zora xe	48	.94	90	.97	92	.97	176	.93
	rhonuc co	283	.99	711	.996	706	.996	977	.99
	135TCC geomopt	461	.99	980	.997	968	.997	1225	.99

(whose nodes sit on the same high-speed local network) and different DAS-2 locations (which are connected through the high-speed SURFnet[20] network), while a degradation in performance, seen in the decrease of percentage of the total time spent on the actual calculations, is visible for the Polish host in the Gridlab testbed. Such degradation is predominantly due to the longer times spent on transferring executables. While this may make the use of this framework impractical for very short calculations (defined as taking less than five minutes on a single CPU), most chemical calculations required for production work take considerably longer, so that it is effective to also use geographically distant hosts. Short jobs that use the same executable could furthermore be combined into a single job, to reduce the transfer/total execution time ratio in that case as well.

4 Conclusions

We have discussed a framework that allows for the use of different quantum chemistry codes in heterogeneous grid environments. This framework is composed of three interacting but formally independent parts, integrated via a command-line interface: the first handling application-specific task of creating an execution script for a given type of calculation; the second handling grid-related tasks, such as submission, execution and retrieval of results; and the third comprising the infrastructure that enables performing calculations in an heterogeneous environment.

The strategy used in creating this framework consists of using components that are available for UNIX-like or Windows-based installations, namely the Python programming Language and the Grid Application Toolkit (GAT). The execution of computational chemistry codes at this time requires the availability of the Bourne shell, and was only tested in UNIX-like systems, but it should be possible to have that available on Windows-based systems, or eventually use the upcoming Microsoft shell (MSH)[21] as a replacement of the Bourne shell in these systems.

In terms of functionality for computational chemistry tasks, the current implementation of this framework provides enough flexibility for the execution of one or more calculations as a single job, irrespective of the codes involved in calculating each step. Given the successful use of this implementation in pilot calculations with the ADF, Dalton, Dirac and Dacapo codes, we are currently working to incorporate other codes into this setup, as well as testing this framework in a production environment.

Acknowledgements

ASPG and LV acknowledge financial support from National Foundation for Computing Facilities (NCF) through the NCF research grant number NRG-2004-05, and technical support from SCM (Scientific Computing and Modeling) via Dr. Stan van Gisbergen and Dr. Olivier Visser regarding different aspects of the use of ADF in this project. Discussions with Dr. Drew McCormack, particularly regarding the use of Apple's Xgrid framework and the setup of the department-wise test grid are also acknowledged. AM acknowledges the support of the Grid-Lab project and the GAT development team, in particular from Hartmut Kaiser, who provided the Python language bindings.

References

1. Greenberg, J.P., Mock, S., Bhatia, K., Katz, M., Bruno, G., Sacerdoti, F., Papadopoulos, P., Baldrige, K.K.: *Future Generation Computer Systems* **21** (2005) 3
2. Sudholt, W., Baldrige, K.K., Abramson, D., Enticott, C., Garic, S., Kondric, C., Nguyen, D.: *Future Generation Computer Systems* **21** (2005) 27
3. Foster, I., Kesselman, C.: *Future Generation Computer Systems* **15** (1999) 607
4. Baldrige, K.K., Greenberg, J.P., Elbert, S.T., Mock, S., Papadopoulos, P.: QMView and GAMESS: Integration into the world wide computational grid. *IEEE* (2002)
5. Nishikawa, T., Nagashima, U., Sekiguchi, S.: ICCS 2003 part III Proceedings Lecture Notes in Computer Science **2659** (2003) 244
6. Kwak, J., Lee, Y.S.: *Journal of Theoretical and Computational Chemistry* **4** (2005) 289
7. Lesyng, B., Bala, P., Erwin, D.: *Journal of Parallel and Distributed Computing* **63** (2003) 590
8. <http://www.gridchem.org>
9. Almond, J., Snelling, D.: *Future Generation Computer Systems* **15** (1999) 539

10. <http://www.scm.com>
11. Allen, G., Davies, K., Goodale, T., Hutanu, A., Kaiser, H., Kielmann, T., Merzky, A., Nieuwpoort, R., Reinefeld, A., Schintke, F., Schutt, T., Seidel, E., Ullmer, B.: The grid application toolkit: Towards generic and easy application programming interfaces for the grid. *Proceedings of the IEEE* **93** (2005) 534
12. K. Keahey and I. Foster and T. Freeman and X. Zhang and D. Galron: Virtual Workspaces in the Grid. In: *Europar 2005*, Lisbon, Portugal (2005)
13. K. Keahey and K. Doering and I. Foster: From Sandbox to Playground: Dynamic Virtual Environments in the Grid. In: *5th International Workshop in Grid Computing (Grid 2004)*, Pittsburgh, PA, USA (2004)
14. <http://www.kjemi.uio.no/software/dalton>
15. <http://dirac.chem.sdu.dk>
16. <http://dcwww.camp.dtu.dk/campos/Dacapo>
17. Bal, H., Bhoedjang, R., Hofman, R., Jacobs, C., Kielmann, T., Maassen, J., Nieuwpoort, R., Romein, J., Renambot, L., Rh, T., an K. Verstoep, R.V., Baggio, A., Ballintijn, G., Kuz, I., Pierre, G., Steen, M., Tanenbaum, A., Doornbos, G., Germans, D., Spoelder, H., Baerends, E.J., Gisbergen, S., Afsermanesh, H., Albada, D., Belloum, A., Dubbeldam, D., Hendrikse, Z., Hertzberger, B., Hoekstra, A., Iskra, K., Kandhai, D., Koelma, D., Linden, F., Overeinder, B., Sloom, P., Spinato, P., Epema, D., Gemund, A., Jonker, P., Radulescu, A., Reeuwijk, C., Sips, H., Knijnenburg, P., Lew, M., Sluiter, F., Wolters, L., Blom, H., Laat, C., Steen, A.: *ACM SIGOPS Operating Systems Review* **34** (2000) 76
18. Seidel, E., Allen, G., Merzky, A., Nabrzyski, J.: *Future Generation Computer Systems* **18** (2002) 1143
19. McCormack, D.: <http://macdevcenter.com/pub/a/mac/2005/08/23/xgrid.html>
20. Tatipamula, M., Bos E. J.: *IEICE Transactions on Communications E878* **3** (2004) 400
21. Nadal, J.: <http://www.developer.com/net/net/article.php/3286851>

Thermal Characteristics and Measurement of Nanoscale Materials

Taikyeong T. Jeong¹ and Young Seok Song^{2,*}

¹ Department of Electrical and Computer Engineering
University of Delaware, Newark, DE 19716 USA
ttjeong@alumni.utexas.net

² Center for Composite Materials
University of Delaware, Newark, DE 19716 USA
youngseoks@gmail.com

Abstract. Numerical prediction of the physical properties of nanocomposites is an attractive area that requires more discussion and investigation. In this paper, we calculate the thermal conductivity of nanocomposites embedded with carbon nanotubes (CNTs) based on the representative volume element (RVE) concept. The RVE, which encompasses a single CNT, was constructed assuming that the CNTs are distributed in polymeric material homogeneously, and also assuming that the CNTs have no interaction with other CNTs. This research describes the thermal characteristics of nanoscale materials - CNTs filled nanocomposites - as a case study and measured their thermal conductivity, for the purpose of validation of numerical results. The dispersion state of the CNTs was observed using field emission scanning electronic microscope (FESEM). We found that the numerically predicted thermal conductivity is closely matches the experimental one and that the numerical tool employed in the study is superior to other analytical and numerical methods.

1 Introduction

Polymeric nanocomposites filled with nanoscale particles such as clay, nanofiber, and carbon nanotubes (CNTs) have been vigorously investigated over the past decade and remarkable achievements have been reported [1]. It was recently reported that CNT nanocomposites possess excellent thermal conductivities which can be applied to production of miniature devices managing heat transfer. In this paper, we focus on the thermal characteristics of the CNT filled polymeric composites but not thermal conductivity of the CNT itself. Most studies on thermal conductivity of CNT nanocomposites have been carried out experimentally or analytically [2]. To consider the contribution of various determinants like CNT geometry or concentration on the thermal conductivity, numerical analysis and calculation is required. In the current study, the effective thermal conductivity tensor of CNT filled composites is discussed using a homogenization method. In

* Manuscript Dec. 12, 2005; T. Jeong was with the University of Texas at Austin and the corresponding author is Y. Song.

particular, the influence of the aspect ratio and the content of CNTs are investigated. In contrast to most of the previous studies on the homogenization method using the FEM, control volume finite element method (CVFEM) is employed in the implementation of the homogenization method. The bounding approach is also adopted to take the orientation state of the CNTs into account and the analytic model [3] is used in the comparison with numerical results. The predicted thermal conductivities of nanoscale materials are compared with experimental results.

2 Computation Method

The AEH method is superior in several ways to other analytic and numerical methods. The contribution of complex geometries and anisotropic material properties of fillers can be precisely calculated through the AEH method as well as both localization and homogenization for heterogeneous materials. In the current study, two different packing structure of the CNTs, square and hexagonal packing, are taken into account so as to evaluate the contribution of CNT packing configuration as shown in Fig. 1 (a) and (b), respectively.

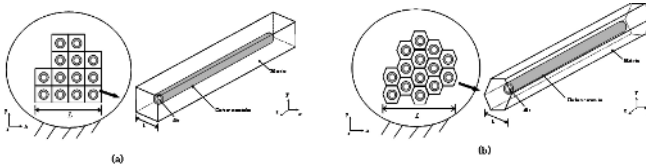


Fig. 1. Schematic illustration of (a) square packed RVE and (b) hexagonally packed RVE in CNT filled nanocomposites

The periodic unit cell consists of three different regions, i.e., matrix, CNT, and air which are denoted by Ω_m , Ω_c , and Ω_a , respectively. Each region has its own thermal conductivity. This study assume the following: (i) CNTs are homogeneously dispersed in the CNT/polymer composites and have uniform dimensions including their length, inner, and outer diameters. (ii) there is no direct interaction between the adjacent CNTs. (iii) the CNT composites contain the periodic unit cell which includes a single CNT embedded unidirectionally. The scale parameter, $\varepsilon = \frac{l}{L} \ll 1$, is defined as the ratio of the characteristic length scales, which implies that the entire dimension, Ω , of RVE is negligible compared to the characteristic length, L , for all of the composites. The temperature field is asymptotically expanded as below.

$$T(x_i, y_i) = T^{(0)}(x_i, y_i) + \varepsilon T^{(1)}(x_i, y_i) + \varepsilon^2 T^{(2)}(x_i, y_i) + \dots \quad (1)$$

where x_i and y_i indicate the global and local length scales. The asymptotic expansion homogenization method (AEH) adopts two distinct coordinate systems, i.e., macroscopic scale x_i and microscopic scale y_i , which can describe the

macroscopic and microscopic behaviors of CNT filled composites. It is assumed that the temperature field is periodic with respect to y_i coordinate system. The global coordinate system and the local coordinate system have the relation of $y_i = \frac{x_i}{\varepsilon}$. The differential operator is written as the following.

$$\frac{\partial}{\partial y_i} = \frac{\partial}{\partial x_i} + \varepsilon \frac{\partial}{\partial x_i} \quad (2)$$

In the steady-state heat conduction problem, the governing equation for each region is shown as below.

$$-\frac{\partial}{\partial x_i} [k_{ij}^m \frac{\partial T}{\partial x_j}] = f_m \quad \text{in } \Omega_m \quad (3)$$

$$-\frac{\partial}{\partial x_i} [k_{ij}^c \frac{\partial T}{\partial x_j}] = f_c \quad \text{in } \Omega_c \quad (4)$$

$$-\frac{\partial}{\partial x_i} [k_{ij}^a \frac{\partial T}{\partial x_j}] = f_a \quad \text{in } \Omega_a \quad (5)$$

where k_{ij}^m , k_{ij}^c , and k_{ij}^a are the second order thermal conductivity tensors for matrix, CNT, and air, respectively. f_m , f_c , and f_a denote the volumetric heat generation for each region. In this study, the heat generation is not considered and the thermal conductivity tensor is assumed to be independent of temperature. After substituting the equation (1) into the equations (3)-(5), the following equations are obtained by collecting the terms with the same order of ε .

For 1th order of ε ,

$$-\frac{\partial}{\partial y_i} [k_{ij}^m \frac{\partial T^1}{\partial y_j}] - \frac{\partial}{\partial y_i} [k_{ij}^m \frac{\partial T^0}{\partial x_j}] - \frac{\partial}{\partial x_i} [k_{ij}^m \frac{\partial T^0}{\partial y_j}] \quad \text{in } \Omega_m \quad (6)$$

$$-\frac{\partial}{\partial y_i} [k_{ij}^c \frac{\partial T^1}{\partial y_j}] - \frac{\partial}{\partial y_i} [k_{ij}^c \frac{\partial T^0}{\partial x_j}] - \frac{\partial}{\partial x_i} [k_{ij}^c \frac{\partial T^0}{\partial y_j}] \quad \text{in } \Omega_c \quad (7)$$

$$-\frac{\partial}{\partial y_i} [k_{ij}^a \frac{\partial T^1}{\partial y_j}] - \frac{\partial}{\partial y_i} [k_{ij}^a \frac{\partial T^0}{\partial x_j}] - \frac{\partial}{\partial x_i} [k_{ij}^a \frac{\partial T^0}{\partial y_j}] \quad \text{in } \Omega_a \quad (8)$$

The perturbation temperature, T^1 , is assumed to be

$$T^1(x_i, y_i) = -\chi_j(y_i) \frac{\partial T^0}{\partial x_j}(x_i) + \tilde{T}^1(x_i) \quad (9)$$

where χ_j is the arbitrary characteristic function which is independent of x_i coordinate system and has periodicity with respect to y_i coordinate system. As a result, the final governing equations are obtained by substituting the equation (12) into the equations (9)-(11) as below.

$$-\frac{\partial}{\partial x_i} [k_{ik}^m \frac{\partial \chi_j}{\partial y_k}] = \frac{\partial}{\partial y_i} k_{ij}^m \quad \text{in } \Omega_m \quad (10)$$

$$-\frac{\partial}{\partial x_i} [k_{ik}^c \frac{\partial \chi_j}{\partial y_k}] = \frac{\partial}{\partial y_i} k_{ij}^c \quad \text{in } \Omega_c \quad (11)$$

$$-\frac{\partial}{\partial x_i} [k_{ik}^a \frac{\partial \chi_j}{\partial y_k}] = \frac{\partial}{\partial y_i} k_{ij}^a \quad \text{in } \Omega_a \quad (12)$$

The homogenized effective conductivity tensor is calculated by integrating the conductivity tensor over the whole domain, Ω , as given by the following equation.

$$\langle k_{ij} \rangle = \frac{1}{\Omega} \int_{\Omega} [k_{ij} - k_{ik} \frac{\partial \chi_j}{\partial y_k}] dy \quad (13)$$

3 Numerical Implementation

The control volume finite element method (CVFEM), which is known as a very efficient numerical method for heat transfer problems is employed as numerical implementation method in this study. Equations (13)-(15) are rewritten in the general form as below.

$$\frac{\partial J_i}{\partial y_i} = S \quad (14)$$

where J_i is the diffusion flux and S is the source term. In the matrix region, Ω_m , the diffusion flux and the source term are given by

$$J_i = -k_{ik}^m \frac{\partial \chi_j}{\partial y_k} + k_{ik}^m \quad (15)$$

$$S = 0 \quad (16)$$

The diffusion flux and the source term for the other regions are obtained in ways similar to the above equations. Because CVFEM is based on the conservation principle within the control volume, we integrate equation (17) over the control volume as the following equation.

$$\int_{\partial V} J_i n_i ds = \int_{\partial V} S dV \quad (17)$$

where ∂V is the surface of control volume, and n_i is the outward unit vector normal to a differential area, ds . In order to integrate equation (20), specification of the interpolation function for the dependent variable, χ , is needed in each element. The linear interpolation function is employed for the characteristic function in the current study. Substitution of the interpolation function into equation (20) yields the complete set of discretization equations. More details on the numerical analysis procedure were presented in the previous study.

The unit cells with unidirectionally aligned CNTs have transversely isotropic physical properties, but the CNTs are dispersed randomly in the composites. Therefore, the effective thermal conductivity for a composite filled with randomly dispersed CNTs is obtained from an orientation averaging procedure. In order to describe the orientation state of CNTs, the second order orientation tensor is adopted. In addition, to validate the numerically predicted thermal conductivities, they are compared with the results obtained by the analytic model proposed by Lewis and Nielsen [3]. This model can explain the effect of the shape and the packing type of the particles for two phase systems. However, it does not take into account the anisotropy of thermal conductivity for CNTs and the contribution of air within the CNT.

4 Measurement

In order to obtain good and homogeneous dispersion of the CNTs in the epoxy resin, acid treatment on the CNTs was carried out, which also removed impurities including amorphous carbons, graphite particles, and metal catalysts. The CNTs were treated in a 3-to-1 mixture of 65% H₂SO₄/HNO₃, under the condition of 100 C for 30 min., after which they were washed with distilled water and then dried in a vacuum oven. The treated CNTs were then dispersed in a solvent and mixed with the epoxy resin. After evaporating the solvent, the hardener was added to the mixture with the CNT loading of 0.5 and 1.0 wt.%. The epoxy and CNT mixture was cast into a cylindrical cavity and cured in a vacuum oven. The specimen had a diameter of 10mm and length of 20mm. Therefore, we measured the thermal conductivity measurement by employing ASTM E1225-87, which was based on temperature difference between the reference and the specimen at steady state and room temperature. For verification of uniform dispersion of the CNTs, morphological characterization was performed by using an FESEM.

5 Results and Discussion

Fig. 2 presents the increase of thermal conductivity of nanocomposites filled with 0.5 wt. % CNTs with respect to the CNT aspect ratio. When the CNTs are hexagonally arranged in the nanocomposites, the nanocomposites have higher thermal conductivities. As the aspect ratio of CNTs increases, thermal conductivities of the nanocomposites with hexagonal or square packing of the CNTs, eventually approaching a limiting value, which can be determined by the rule for mixture of conductivities for distinct regions. From here, the thermal conductivities predicted numerically mean the values calculated for the square packing of the CNT.

In Fig. 3, the thermal conductivities calculated by the analytic model and the homogenization technique are compared in the axial and the parallel directions of the CNT when the CNT loading is 0.5 wt. % (0.29 % volume fraction). In order to vary the aspect ratio of CNTs, length of the CNT in the RVE is altered. In the case of the axial direction of the CNT, both analytic and numerical predictions

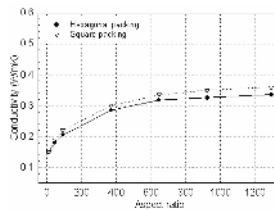


Fig. 2. Effect of packing the configuration of the CNTs' thermal conductivities in the axial direction when the CNTs of 0.5 wt. % are filled in nanocomposites

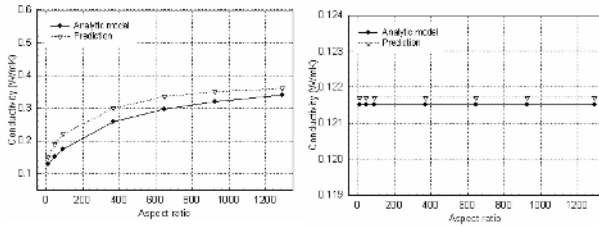


Fig. 3. Effective thermal conductivities with respect to aspect ratio of CNT (a) in the axial direction and (b) in the parallel directions

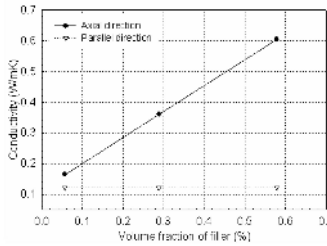


Fig. 4. Effective thermal conductivities calculated numerically with respect to CNT volume fraction

are increased with on increase in the aspect ratio. The thermal conductivity obtained by the homogenization method increases more rapidly with respect to the aspect ratio than that obtained the analytic model. When the CNT length is relatively large, the homogenization method is closer to material properties averaged by the rule of mixture with the volume fraction. On the other hand, it is shown in Fig. 3 (b) that thermal conductivities in the parallel direction of the CNTs have little variation with respect to the aspect ratio of CNT. The effect of the CNT volume fraction on the thermal conductivity is examined as shown in Fig. 4 and Fig. 5. Therefore, Fig. 4 shows that the thermal conductivities of the nanocomposites in the axial direction are much higher compared to the parallel direction and that the difference between them becomes large as the volume fraction of the CNT increase.

In addition, Fig. 5 (a) presents the effective thermal conductivity of composites filled with randomly oriented CNTs. The experimental measurement was carried out for CNT volume fractions of 0.29 and 0.58 %. The numerical and experimental data for higher volume fraction of CNTs are slightly different. The discrepancy has two causes. First, in the case that the volume fraction is relatively high, it is difficult to reach the homogeneous dispersion without aggregation of the CNTs, as was assumed for this study. Second, the bounding approach using the orientation tensor cannot completely take into account the contribution of the randomly oriented CNTs. Fig. 5 (b) shows an FESEM image of the composite filled with CNTs of 0.5 wt. %. The image indicates that the embedded CNTs are homogeneously dispersed in the epoxy resin and direct contact between

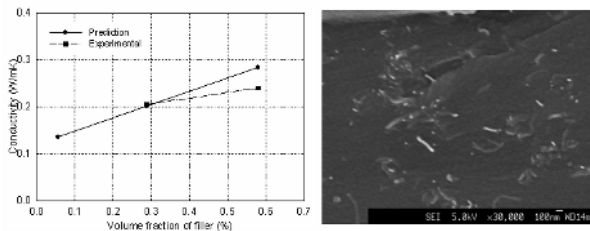


Fig. 5. (a) Effective thermal conductivity for the composites filled with randomly dispersed CNTs with respect to CNT volume fraction and (b) morphological observation of CNTs by FESEM

neighboring CNTs is negligible. This figure confirms the assumption that there is no direct interaction between the adjacent CNTs. Although the CNT loading is too small to reach the electrical percolation, mechanical and rheological properties of CNT filled composites can be enhanced dramatically [1]. On the other hand, it has been shown that thermal conductivities do not show the percolation behavior in contrast to electrical conductivities [6].

6 Conclusion

The current study shows that the homogenization technique is a promising tool because it can consider such anisotropic material properties as thermal conductivity, elastic modulus, and thermal expansion coefficient as well as geometrical complexity. Specifically speaking, the effective thermal conductivity of the composites filled with CNTs is evaluated by the homogenization technique. The control volume finite element method (CVFEM) is employed so as to implement the homogenization method. The two different RVE structures filled with a single CNT are adopted in calculation of the thermal conductivity of composites. It is assumed that the composites have geometric periodicity with respect to local length scale. Orientation of the CNTs embedded in the composites is taken into account using the bounding approach. The effect of the aspect ratio and volume fraction of CNTs on the effective thermal conductivity was investigated. The numerically calculated conductivities closely match the experimental result.

References

1. C. A. Mitchell, J. L. Bahr, S. Arepalli, J. M. Tour, R. Krishnamoorti, "Dispersion of functionalized carbon nanotubes in polystyrene," *Macromolecules*, vol. 35, pp. 8825-8830, 2002
2. P. Chantrenne, J. L. Barrat, "Analytical model for the thermal conductivity of nanostructures," *Suprelattice and Microstructure*, vol. 35, pp. 173-186, 2004
3. T. B. Lewis, L. E. Nielsen, "Dynamic mechanical properties of particulate-filled composites," *J. Applied Polymer Science*, vol. 14, pp. 1449-1471, 1970

4. Y. S. Song, K. Chung, T. J. Kang, J. R. Youn, "Prediction of permeability tensor for plain woven fabric by using control volume finite element method," *Polymers and Polymer Composites*, vol. 11, no. 6, pp. 465-476, 2003
5. S. G. Advani, C. L. Tucker III, "The use of tensor to describe and predict fiber orientation in short fiber composites," *J. Rheol.*, vol. 31, no. 8, pp. 751-784, 1987
6. Y. P. Manunya, V. V. Davydenko, P. Pissis, E. V. Lebedev, "Electrical and thermal conductivity of polymers filled with metal powders," *European Polym. J.*, vol. 38, pp. 1887-1897, 2002
7. T. Jeong and A. Ambler, "Power efficiency system for flighth application (PESFA) mission: Low power dissipation in digital circuit design for flighth application/space communications," *IEEE Tran. on Aerospace and Electronics System*, ISSN: 0018-9251, 2006

Computational Analysis and Simulation of Vacuum Infusion Molding Process

Young Seok Song¹ and Taikyeong T. Jeong^{2,*}

¹ Center for Composite Materials
University of Delaware, Newark, DE 19716 USA
youngseoks@gmail.com

² Department of Electrical and Computer Engineering
University of Delaware, Newark, DE 19716 USA
ttjeong@alumni.utexas.net

Abstract. The current work focuses on resin bleeding process during the Seeman composite resin infusion molding process (SCRIMP), which is a subset of liquid composite molding (LCM) process. Finite difference method (FDM) is implemented to predict the preform thickness, bleeding resin volume, and fiber volume fraction by using a non-rigid control volume. After the fibrous preform is completely impregnated, the resin flow within the preform has a great impact on the dimension and mechanical properties of the final composite parts. As the resin flows out of the preform, the resin pressure and preform thickness are reduced, which increases the fiber volume fraction and the dimension tolerance of the preform.

In this paper, the influence of resin flow rate at vent in the mold is also investigated. It is found that there is a critical flow rate to optimize the SCRIMP process at the vacuum line.

1 Introduction

The SCRIMP is a variant of the traditional resin transfer molding (RTM) process, which is known to be suitable for processing large-scale composite structures [1], [2], [3]. The process can be explained as follows. The dry preform is laid up and covered with a very high permeable media. A peel ply is laid in between them. All of the layers are then sealed in a plastic film/bag and then vacuum is applied to evacuate entrapped air. When the resin inlet line is opened the resin is driven by atmosphere pressure through the infusion line until the preform is fully impregnated and the resin is cured. Since the process uses a single-sided tool, room temperature resin and vacuum only, SCRIMP shows great potential ability to form large-scale composites parts [4], [5], [6].

The resin bleeding stage of SCRIMP is from the time when preform is fully infused to the time when resin is completely cured. The flow behavior of resin during this period will significantly affect the dimensional tolerance, fiber volume fraction and the void content of the composite. Spatial variation in thickness

* T. Jeong is the corresponding author.

and fiber volume fraction influences the mechanical properties of the composite laminate. In the SCRIMP process, resin is infused by opening the infusion line. During the infusion process, there exists a pressure gradient along the preform from the infusion line to the vacuum line. Since the sealed preform is not confined by the mold on top, the resin pressure gradient should cause a thickness gradient of the preform, which will be frozen in the cured final composite part. SCRIMP processing can be described as two different stages. The first stage is the infusion stage, which ends when the preform is infused completely. The second stage is the post-infusion stage, which starts after the first stage and ends when the resin is completely cured. Much attention has been paid on the first stage and much research has been done [6], [7], [8], [9] [10], to understand the resin flow and saturation behavior and to fulfill a proper infusion strategy. When the infusion is complete, therefore, one generally closes the infusion line or keeps resin circulating till the resin cures. It seems that the entire SCRIMP process has been complete by the end of the first stage, however, during the post-infusion stage, resin flow, pressure gradient, thickness gradient and the rheological properties of the resin continue changing as a function of time and location. These changes will finally determine the dimensional tolerance, fiber volume fraction; void content and mechanical properties of the composite.

In this paper, SCRIMP process is investigated by using FDM. We, particularly, establish a computational model for resin bleeding process in SCRIMP to analyze the resin flow behavior and examine the effects of different processing parameters, resin and fabric properties on the change of laminate thickness and fiber volume fraction.

2 Description of Resin Bleeding Process

2.1 Governing Equations

Darcy's law can be used as the basic momentum governing equation. For one-dimensional problem, it can be expressed as follows.

$$\mu_x = \frac{Q_x}{A} = -\frac{K}{\mu} \cdot \frac{dP}{dx} \quad (1)$$

where μ_x is the flow velocity in x-axes and A is the cross-sectional area of preform. K is the effective permeability of the preform, μ is the viscosity of resin, and $\frac{dP}{dx}$ is the pressure gradient of the resin. When the filled mold is rigid, the one-dimensional equation of mass conservation for the fluid can be written as.

$$\frac{d\mu_x}{dx} = 0 \quad (2)$$

Substituting equation (1) into equation (2) leads to:

$$\frac{d}{dx} \left(\frac{K}{\mu} \cdot \frac{dP}{dx} \right) = 0 \quad (3)$$

This equation implies a linear pressure distribution between the resin infusion line and the flow front. If the resin pressure at the infusion line is P_0 , the pressure distribution would have of the following form.

$$P = P_0 \left(1 - \frac{x}{x_f}\right) = 0 \quad (4)$$

where x_f is the location of the flow front.

During the SCRIMP process, the preform is compressed by the atmospheric pressure and there is no rigid mold on top of the preform. Therefore the resin pressure gradient will cause the thickness variation across the preform. Since the preform for SCRIMP is not rigid, the resin pressure distribution may deviate from equation (4). However, when the preform is fully infused, the thickness will not change if one circulates the resin. At that situation, the preform can be approximately considered rigid and the resin pressure can be expressed by equation (4).

The flow behavior of resin and thickness change during the post-infusion period is related to different processing scenarios and parameters. After the preform is infused completely, if one circulates the resin until cured, a pressure gradient in the part will remain the same as that during the infusion period and thus the thickness gradient will be frozen into the part after cure. So, measures usually need to be taken before the resin is cured.

SCRIMP process can be assumed as compressible in the thickness direction and incompressible in the in-plane directions. A one-dimensional semi-rigid control volume of the preform after complete infusion is employed in the current study. We obtain equation (5), from the mass conservation principle.

$$Q_{in} - Q_{out} = dv \quad (5)$$

where Q_{in} is the flow rate entering the control volume and Q_{out} is the flow rate out of the control volume. dv denotes the rate of volume change of the control volume. By using Darcy's law, flow rates in and out to the control volume can be expressed as.

$$Q_{in} = -\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} A_{in} = -\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} W \varepsilon(x, t) \quad (6)$$

$$Q_{out} = -\frac{K}{\mu} \frac{\partial P(x + dx, t)}{\partial x} A_{out} = -\frac{K}{\mu} \frac{\partial P(x + dx, t)}{\partial x} W \varepsilon(x + dx, t) \quad (7)$$

$$\begin{aligned} &\approx -\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} W \varepsilon(x, t) + \frac{\partial}{\partial x} \left[-\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} W \cdot \varepsilon(x, t) \right] \cdot dx \\ dv &= \frac{\partial \varepsilon(x, t)}{\partial t} \cdot W \cdot \varepsilon(x, t) \end{aligned} \quad (8)$$

where W is the width, ε is the thickness of preform, and t is the time. After substituting equation (6), (7), and (8) into equation (5), we finally obtain

$$\frac{\partial}{\partial x} \left[-\frac{K}{\mu} \cdot \frac{\partial P(x, t)}{\partial x} \cdot W \cdot \varepsilon(x, t) \right] = \frac{\partial \varepsilon(x, t)}{\partial t} \quad (9)$$

The thickness of the preform at any time can be expressed as

$$\varepsilon(x, t) = f(P(x, t)) \tag{10}$$

Researchers have modeled the relation between preform thickness and resin pressure both empirically and analytically. We have used a fourth order polynomial empirical equation as our compaction model. Equation (9) can then be rewritten as

$$\frac{\partial}{\partial x} \left[-\frac{K}{\mu} \cdot \frac{\partial P(x, t)}{\partial x} \cdot W \cdot f(P(x, t)) \right] = \frac{\partial f(P(x, t))}{\partial t} \tag{11}$$

2.2 Computational Scheme

When the infusion line is closed, there is no resin entering into the preform. At the boundary surface $x = 0$, the flow rate entering a control volume $Q_{in} = 0$. The flow rate of resin out of the control volume is equal to the rate of change of volume.

$$\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} \cdot \varepsilon(x, t) = \frac{\varepsilon(x, t)}{\partial t} \cdot dx \tag{12}$$

On the other end, $x = L$, resin flows out through a tube. Considering the control volume at the end, the flow rate into the control volume is

$$Q_{in} = -\frac{K}{\mu} \cdot \frac{\partial P(x, t)}{\partial x} \cdot W \cdot \varepsilon(x, t) \tag{13}$$

The flow rate out of the volume is actually the flow rate in the tube and can be expressed as

$$Q_{out} = Q_{tube} = \frac{\pi \cdot a^4}{8\mu \cdot L_0} \cdot P(x, t) \tag{14}$$

where a is the radius of the tube, L_0 is the length of the tube, and μ is the viscosity of the resin. Therefore the boundary condition can be expressed as,

$$-\frac{K}{\mu} \frac{\partial P(x, t)}{\partial x} \cdot \varepsilon(x, t) - \frac{HP(x, t)}{W} = \frac{\partial \varepsilon(x, t)}{\partial t} \cdot dx \tag{15}$$

where $H = \frac{\pi \cdot a^4}{8\mu \cdot L_0}$. At the moment when infusion line is closed ($t = 0$), the resin pressure can be derived from the one dimensional Darcy’s law and conservation principle. It is well known that the permeability of the preform is determined by the structure and fiber volume fraction of the preform. Since the thickness of the preform changes with time and location during the process, therefore, the permeability is also a function of time and location. The following equation is the relationship between permeability and fiber volume fraction.

$$K = \kappa \frac{(1 - V_f)^3}{V_f^2} \tag{16}$$

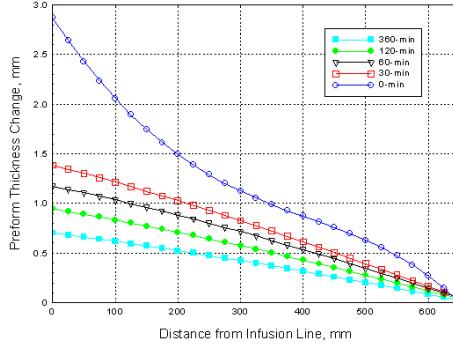


Fig. 1. Preform thickness change as a function of location (vacuum tube radius $a = 2$ mm)

where κ is the Kozeny Carman constant and V_f is the fiber volume fraction. One-dimensional central finite difference method was employed to implement the numerical analysis. The detail description of finite difference scheme is shown in Figure 1. The length of preform was divided into m segments and time was also divided into small steps. Using the difference scheme presented, equation (9) can be discretized as

$$\frac{\partial}{\partial x} \left[\frac{K}{\mu} \cdot \frac{\partial P(x, t)}{\partial x} \cdot \varepsilon(x, t) \right] \approx \frac{K_i^n}{\mu^n} \cdot \left(\frac{\varepsilon_i^n \cdot \frac{P_{i+1}^n - P_i^n \Delta x}{\Delta x} - \varepsilon_{i-1}^n \cdot \frac{P_i^n - P_{i-1}^n \Delta x}{\Delta x}}{\Delta x} \right) \quad (17)$$

$$\frac{\partial \varepsilon(x, t)}{\partial t} = \frac{\varepsilon_i^n - \varepsilon_{i-1}^n}{\Delta t} \quad (18)$$

Then, equation (9) can be recast as

$$\varepsilon_i^n + 1 = r_i^n \cdot [\varepsilon_i^n \cdot (P_{i+1}^n - P_i^n) - \varepsilon_{i-1}^n \cdot (P_i^n - P_{i-1}^n)] + \varepsilon_i^n \quad (19)$$

where $r_i^n = \frac{K_i^n}{\mu^n} \cdot \frac{\Delta t}{\Delta x^2}$.

3 Results and Discussion

Figure 1 shows the preform thickness changes as a function of distance from infusion line at different time during the post-infusion period in the case of vacuum tube radius, a , of 2 mm. The origin point is the infusion line position and the point $x = 650$ mm is the position of the vacuum line. It can be seen that the thickness at the infusion side decreased and that at vacuum side increased with time. As time passed by, the thickness gradient became smaller and finally became almost uniform after 6 hours when the gel point of the resin is reached. In the case of the tube radius is larger than 2 mm, the thickness at vacuum side not changed with time. The decrease of thickness gradient was only caused by the decrease of thickness at the infusion side.

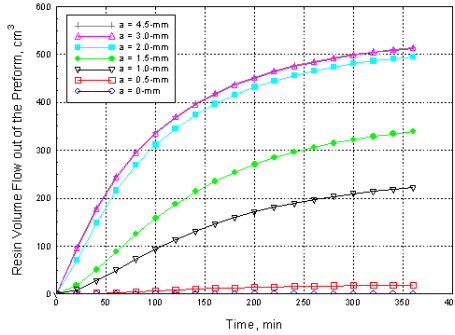


Fig. 2. Resin volume flowing out of the preform as a function of time with respect to various tube diameters, from 0 mm to 4.5 mm

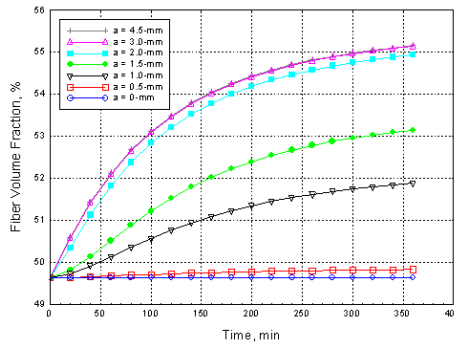


Fig. 3. Fiber volume fraction as a function of time with respect to various tube diameters, from 0 mm to 4.5 mm

Figure 2 presents the resin volume flowing out of preform during post-infusion and Figure 3 is the fiber volume fraction change with time when the various tube diameters (from 0 mm to 4.5 mm) are applied. It can be seen that larger vacuum tube sucked out more resin from the perform and the panel reached higher fiber volume fraction. However, excessive loss of resin before gel point may lead to voids in the composite laminate. Uniform thickness, high fiber volume fraction and low void content are the objectives for SCRIMP processing. Using the analysis presented above, we can balance and optimize these properties by choosing the proper radius of vacuum tube.

4 Conclusion

The post-infusion stage of the SCRIMP process significantly affects the dimensional tolerance, fiber volume fraction, and the bleeding resin volume. FDM is implemented to predict the preform thickness, bleeding resin volume, and fiber

volume fraction by using a non-rigid control volume. As the resin flows out of the preform, the resin pressure and preform thickness are reduced, which increases the fiber volume fraction and the dimension tolerance of the preform. In this paper, the influence of resin flow rate at vent in the mold is also discussed. This work provides a practical tool to optimize the SCRIMP processing and the final composite properties.

Acknowledgment

The author would like to thank the referees for their valuable comments.

References

1. S. M. Lewit, J. C. Jakubowski, "Low cost VARTM process for commercial and military applications," *42nd International SAMPE Symposium*, vol. 42, pp. 1173-1187, May 4-8, 1997
2. L. Joubaud, V. Achim, F. Trochu, "Numerical simulation of resin infusion and reinforcement consolidation under flexible cover," *Polym. Compos.*, vol. 26, pp. 417-427, 2005
3. Y. S. Song, K. Chung, T. J. Kang, J. R. Youn, "Prediction of permeability tensor for three dimensional circular braided preform by applying a finite volume method to a unit cell," *Compos. Sci. Technol.*, vol. 64, pp. 1629-1636, 2004
4. S. W. Beckwith, C. R. Hyland, "Resin transfer molding: a decade of technology advances," *SAMPE Journal*, vol. 34, pp. 7-19, 1998
5. M. K. Kang, W. I. Lee, H. T. Hahn, "Analysis of vacuum bag resin transfer molding process," *Phys. Lett. A*, vol. 329, pp. 207-213, 2004
6. T. G. Gotowski, G. Dillon, "The elastic deformation of fiber bundles, Advanced composite manufacturing," *New York John Wiley & Sons*, 1997
7. R. Mathur, D. Heider, C. Hoffmann, J. W. Gillespie, S. G. Advani, B. K. Fink, "Flow front measurements and model validation in the vacuum assisted resin transfer molding process," *Polym. Compos.*, vol. 22, pp. 477-490, 2001
8. Y. Song, W. Chui, J. Glimm, B. Lindquist, F. Tangerman, "Applications of front tracking to the simulation of resin transfer molding," *Computers Math. Applic.*, vol. 33, pp. 47, 1997
9. T. Jeong and A. Ambler, "Design trade-offs and power reduction techniques for high performance circuits and system," *Springer-Verlag*, ISSN: 0302-9743, 2006
10. J. Ni, Y. Zhao, L. J. Lee, S. Nakamura "Analysis of two-regional flow in liquid composite molding," *Polym. Compos.*, vol. 18, pp. 254, 1997

Forward, Tangent Linear, and Adjoint Runge-Kutta Methods in KPP–2.2

Philipp Miehe and Adrian Sandu

Department of Computer Science, Virginia Polytechnic Institute and State
University, Blacksburg, VA 24061
{pmiehe, asandu}@cs.vt.edu

Abstract. This paper presents the new stiff solvers of the new version 2.2 of the Kinetic PreProcessor (KPP). Taking a set of chemical reactions and their rate coefficients as input, KPP generates Fortran90, Fortran77, Matlab, or C code for the temporal integration of the kinetic system. Efficiency is obtained by carefully exploiting the sparsity structures of the Jacobian and of the Hessian. A set of integration methods was added to the comprehensive suite of stiff numerical integrators. Moreover, KPP is now ready to be used to generate the tangent linear model, as well as the continuous and discrete adjoint models of the chemical system to do sensitivity analysis.

1 Introduction

The application of computer modeling in atmospheric chemistry requires efficient tools for simulation and analysis of chemical reaction mechanisms. Due to the side by side existence of very stable (e.g., CH_4) and very reactive (e.g., O^{1D}) species numerically challenging software has been developed to integrate stiff ordinary differential equations (ODEs), e.g., Facsimile [6], AutoChem [19], Spack [7], Chemkin [20], Odepack [21], and KPP [1].

Many academic, research, and industry groups in different countries are currently using KPP [8, 9, 11, 12, 10]. The Master Chemical Mechanism (MCM [22]) now provides the option for output in KPP syntax. In this paper we focus on new integrators introduced in version 2.2 of KPP. These integrators allow for high accuracy in efficient simulation of chemical kinetic systems in Fortran90 and Matlab.

The paper is organized as follows. A short overview of usage for KPP is given. The new integrators are presented in Sec. 3. Here we show the mathematical background and mention implementation details. Section 4 shows results from applying the solvers to a chemical system. The conclusion finally is presented in Sec. 5.

2 The Kinetic PreProcessor KPP

The Kinetic PreProcessor KPP [1, 2, 3] is a software tool that assists the computer simulation of chemical kinetic systems. The concentrations of a chemical

system evolve in time according to the differential law of mass action kinetics. A numerical simulation requires an implementation of the differential laws and a numerical integration in time. KPP is currently being used by many academic, research, and industry groups in several countries [8, 9, 11, 12, 10]. The well-established Master Chemical Mechanism (MCM [22]) has also recently been modified to add the option of producing output in KPP syntax.

KPP translates a specification of the chemical mechanism into Fortran77, Fortran90, C, or Matlab simulation code that implements the concentration time derivative function, its Jacobian, and its Hessian, together with a suitable numerical integration scheme. Sparsity in Jacobian/Hessian is carefully exploited in order to obtain computational efficiency. Fortran90 is the programming language of choice for the vast majority of scientific applications. Matlab [25] provides a high-level programming environment for algorithm development, numerical computations, and data analysis and visualization. The Matlab code produced by KPP allows a rapid implementation and analysis of a specific chemical mechanism.

KPP incorporates a library with several widely used atmospheric chemistry mechanisms; the users can add their own chemical mechanisms to the library. KPP also includes a comprehensive suite of stiff numerical integrators. The KPP development environment is designed in a modular fashion and allows for rapid prototyping of new chemical kinetic schemes as well as new numerical integration methods.

A summary of KPP generated routines is given below:

1. *Fun*: the time derivative of concentrations;
2. *Jac*, *Jac_SP*: Jacobian of *Fun* in full or in sparse format;
3. *KppDecomp*: sparse LU decomposition for the Jacobian;
4. *KppSolve*, *KppSolveTR*: solve sparse system with the Jacobian matrix and its transpose;
5. *Jac_SP_Vec*, *JacTR_SP_Vec*: sparse Jacobian (transposed or not) times vector;
6. The stoichiometric matrix *STOICM*;
7. *ReactantProd*: vector of reaction rates;
8. *JacReactantProd*: the Jacobian of the above;
9. *dFun_dRcoeff*: derivatives of *Fun* with respect to reaction coefficients (in sparse format);
10. *dJac_dRcoeff*: derivatives of *Jac* with respect to reaction coefficients times user vector;
11. *Hess*: the Hessian of *Fun*; this 3-tensor is represented in sparse format;
12. *Hess_Vec*, *HessTR_Vec*: Hessian (or its transpose) times user vectors; same as the derivative of Jacobian (transposed) vector product times vector.

3 The Numerical Integrators

The new numerical integrators expand the original set of stiff solvers for ODEs for chemical kinetic systems. The KPP numerical library provides implementations of several stiff numerical solvers. Efficiency is obtained through the use

of sparse linear algebra routines generated by KPP. Several Rosenbrock methods of various orders are implemented in KPP [4]. These methods have proved to be very efficient on many applications, and especially in atmospheric chemistry. The variable order stiff extrapolation code SEULEX [14] is able to produce highly accurate solutions. The Livermore ODE solver (LSODE, LSODES [16]) implements backward differentiation formula (BDF) methods for stiff problems. The solver VODE [17] uses a different formulation of backward differentiation formulas. The BDF-based direct-decoupled sensitivity integrator ODESSA [15] has been modified to use the KPP sparse linear algebra routines.

In this paper we focus on the new implementations of stiff Runge-Kutta methods which have been added to the KPP library. A general s -stage Runge-Kutta method is defined as [13]

$$y^{n+1} = y^n + h \sum_{i=1}^s b_i k_i, \quad T_i = t^n + c_i h, \quad Y_i = y^n + h \sum_{j=1}^s a_{ij} k_j, \quad (1)$$

$$k_i = f(T_i, Y_i),$$

where the coefficients a_{ij} , b_i and c_i are prescribed for the desired accuracy and stability properties. The stage derivative values k_i are defined implicitly, and require solving a (set of) nonlinear system(s). Singly diagonally-implicit Runge-Kutta (SDIRK) methods [13] are defined by (1) with the coefficients $a_{ii} = \gamma$ and $a_{i,j} = 0$ for all i and $j > i$.

Two families of solvers have been added to the KPP library, with multiples methods within each family. These methods are discussed next.

Fully implicit Runge-Kutta methods. The three stage Radau-2A is a Runge-Kutta method of order 5 based on Radau-IIA quadrature and stiffly accurate. While Radau-2A is relatively expensive (when compared to the Rosenbrock methods), it is more robust and is useful to obtain accurate reference solutions. The discrete adjoint of Radau-2A methods are the Radau-1A methods with the same number of stages; therefore the discrete adjoints of Radau-2A methods are not stiffly accurate. The three stage Lobatto-3C method of order 4 is another stiffly accurate method, whose discrete adjoint is the Lobatto-3C method

Table 1. The coefficients of Radau-2A method (left) of order 5 and of the Lobatto-3C method (right) of order 4

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$	0	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{6}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$	1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 2. The coefficients of Sdirk-2a (left), Sdirk-2b (middle), and Sdirk-3a (right) methods. All methods have order 2.

$\frac{2-\sqrt{2}}{2} \mid \frac{2-\sqrt{2}}{2} \quad 0$	$\frac{2+\sqrt{2}}{2} \mid \frac{2+\sqrt{2}}{2} \quad 0$	$\frac{3-\sqrt{3}}{6} \mid \frac{3-\sqrt{3}}{6} \quad 0 \quad 0$
$1 \quad \frac{\sqrt{2}}{2} \quad \frac{2-\sqrt{2}}{2}$	$1 \quad -\frac{\sqrt{2}}{2} \quad \frac{2+\sqrt{2}}{2}$	$1 - \frac{\sqrt{3}}{3} \quad \frac{3-\sqrt{3}}{6} \quad \frac{3-\sqrt{3}}{6} \quad 0$
$\frac{\sqrt{2}}{2} \quad \frac{2-\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2} \quad \frac{2+\sqrt{2}}{2}$	$1 \quad \frac{3-\sqrt{3}}{6} \quad \frac{\sqrt{3}}{3} \quad \frac{3-\sqrt{3}}{6}$
		$\frac{3-\sqrt{3}}{6} \quad \frac{\sqrt{3}}{3} \quad \frac{3-\sqrt{3}}{6}$

itself. Thus the discrete adjoint is stiffly accurate. The Butcher tableaux for these methods are presented in table 1.

Singly-diagonally-implicit Runge-Kutta methods. Sdirk-2 are 2-stage, L-stable, stiffly accurate methods of order 2. The choice $\gamma = 1 - \sqrt{2}/2$ (Sdirk-2a) is more accurate, while the choice $\gamma = 1 + \sqrt{2}/2$ (Sdirk-2b) may be advantageous when a non-negative numerical solution (concentrations) are needed.

Sdirk-3a is a 3-stage, second order, stiffly accurate method. Its coefficients are chosen such that the discrete adjoint is also stiffly accurate.

The methods Sdirk-4 are the fourth order L-stable singly-diagonally-implicit Runge-Kutta methods developed by Hairer and Wanner [14]. Specifically, Sdirk-4a is the method with $\gamma = 4/15$ and Sdirk-4b the method with $\gamma = 1/4$. The coefficients of these methods are given in [14] and not reproduced here.

Implementation Aspects. Following [14, Section IV.8], for implementation purposes (1) is written in terms of the variables $Z_i = Y_i - y^n$. Replacing the nonlinear system in k_i by a nonlinear system in Z_i has numerical advantages for stiff systems where f has a large Lipschitz constant. With the compact notation

$$Z = [Z_1 \cdots Z_s]^T, \quad F(Z) = [f(T_1, y^n + Z_1) \cdots f(T_s, y^n + Z_s)]^T,$$

the nonlinear system (1) in Z is

$$Z = (hA \otimes I_n) \cdot F(Z), \tag{2}$$

where \otimes denotes the Kronecker product. This system can be solved by simplified Newton iterations of the form

$$\begin{aligned} [I_{ns} - hA \otimes J(t^n, y^n)] \cdot \Delta Z^{[m]} &= Z^{[m]} - (hA \otimes I) F^{[m]} \\ Z^{[m+1]} &= Z^{[m]} - \Delta Z^{[m]} \end{aligned} \tag{3}$$

with J the Jacobian of the ODE function. The linear systems in (3) have dimension $ns \times ns$. For SDIRK methods (3) is a sequence of s nonlinear systems of dimension $n \times n$. For fully implicit methods a transformation of the system (3) to complex form is used following [14].

Using the implementations of the forward routines, integrators for the tangent linear model (TLM) and adjoint model (ADJ) were developed. Each family of forward methods (implicit Runge-Kutta, singly-diagonally-implicit Runge-Kutta, and Rosenbrock methods) was extended to calculate the TLM, the generated result is the sensitivity matrix.

Small perturbations of the solution (due to small changes δy^0 in the initial conditions) propagate forward in time according to the *tangent linear model*. The variation of (1) is equivalent to applying the Runge-Kutta to the sensitivity equation to get the tangent linear Runge-Kutta methods:

$$\begin{aligned} y^{n+1} &= y^n + h \sum_{i=1}^s b_i k_i, & \delta y^{n+1} &= \delta y^n + h \sum_{i=1}^s b_i \ell_i, \\ Y_i &= y^n + h \sum_{j=1}^s a_{ij} k_j, & \delta Y_i &= \delta y^n + h \sum_{j=1}^s a_{ij} \ell_j, \\ k_i &= f(T_i, Y_i), & \ell_i &= J(T_i, Y_i) \cdot \delta Y_i. \end{aligned} \quad (4)$$

The system (4) is linear and does not require an iterative procedure. However, even for a SDIRK method ($a_{ij} = 0$ for $i > j$ and $a_{ii} = \gamma$) each stage requires the LU factorization of a different matrix. To avoid repeated factorizations our implementation solves the linear system (4) by iterations of the form (3), re-using the LU decomposition of the matrix based on $J(t^n, y^n)$.

The adjoint methods present the final additions to the library of integrators. By solving the system in forward mode and using checkpoints to save the states at each step of the computation the sensitivity matrix is calculated in backward mode on these checkpoints.

The discrete adjoint of the Runge-Kutta method (1) is

$$\begin{aligned} u_i &= h J^T(T_i, Y_i) \cdot \left(b_i \lambda^{n+1} + \sum_{j=1}^s a_{j,i} u_j \right), & i &= s, \dots, 1 \\ \lambda^n &= \lambda^{n+1} + \sum_{j=1}^s u_j. \end{aligned} \quad (5)$$

For $b_i \neq 0$ the RK adjoint can be rewritten as another Runge-Kutta method. To avoid repeated factorizations our implementation solves the linear system (5) by iterations of the form (3), re-using the LU decomposition of the matrix based on $J(t^n, y^n)$. The forward backward substitutions correspond to the transpose of the system (3).

4 Application and Results

In this section we illustrate the behavior of the new integrators applied to a chemical model. The computations were tested against an independent code: Fortran77 SEULEX stiff differential equations solver by Hairer and Wanner [14].

4.1 The Chemical Model

The chemical model chosen is the SAPRC99 model [18]. It includes 74 species and 211 chemical reaction equations on these species. The tests were performed on data after a 24 hour stabilizing period. The integrators were applied on a 24 hour interval. Emissions have been included in the computations.

4.2 Forward Methods

All new solvers have been tested thoroughly. The forward integrators were tested on the full range of relative error tolerances of 10^{-1} up to 10^{-8} . The absolute error tolerances were adjusted accordingly ($10^3 \cdot \text{rel_tol}$). Results can be seen in Fig. 1. While the implicit Runge-Kutta methods Radau-2A and Lobatto-3C are still very time efficient at high accuracy, the Rosenbrock and Sdirk methods are very efficient in lower accuracy areas. The Rodas-4 integrator shows very powerful and fast performance in our application.

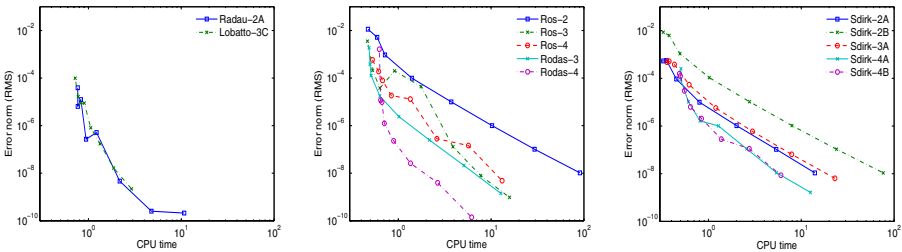


Fig. 1. Comparison of time efficiency and accuracy of forward Runge-Kutta, Rosenbrock and Sdirk methods

4.3 Tangent Linear Methods

Main feature of the new implementation are the sensitivity calculations. The Tangent Linear Model provides the background for the TLM-solvers. The tested relative error tolerances were 10^{-3} , 10^{-6} , 10^{-9} , and 10^{-12} . The default settings of Radau-2A stopped the calculations when applying the tolerance of 10^{-12} - too many steps were used, here we didn't report a result. Figure 2 presents four exemplary results when using the newly implemented integrators. Time-efficiency of Sdirk-4B is the best for any lower accuracy in our model.

4.4 Adjoint Methods

The adjoint methods also provide the sensitivity matrix. The backward scheme requires a large amount of memory storage for results of the forward computation. In order to obtain higher accuracy, even more memory will be used. We have tested our code on error tolerances of 10^{-3} , 10^{-6} , and 10^{-9} . Here, Rodas-4 outperformed Sdirk-4B. In the future we will also implement adjoints based on Radau-2A and Lobatto-3C, because they may achieve higher efficiency on increased accuracy.

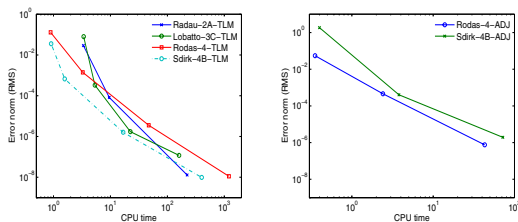


Fig. 2. Comparison of time efficiency and accuracy of tangent linear model (left) and adjoint model (right)

5 Conclusion

State-of-the-art and high-order stiff ODE integrators were added to the library of the widely-used software environment KPP. The new forward methods can achieve a high accuracy. The stability matrix of the species can be computed by using the newly implemented tangent linear model methods as well as adjoint methods. Researchers will now be enabled to easily include stability considerations into their model.

The KPP-2.2 source code is distributed under the provisions of the GNU public license [23] and is available on the web [24].

Acknowledgments

This work was supported by the National Science Foundation through the awards NSF ITR AP&IM 0205198, NSF CAREER ACI0413872, and NSF CCF0515170, by the National Oceanic and Atmospheric Administration (NOAA) and by the Texas Environmental Research Consortium (TERC).

References

1. V. Damian, A. Sandu, M. Damian, F. Potra, and G.R. Carmichael: “The Kinetic PreProcessor KPP – A Software Environment for Solving Chemical Kinetics”, *Computers and Chemical Engineering*, Vol. 26, No. 11, p. 1567–1579, 2002.
2. A. Sandu, D. Daescu, and G.R. Carmichael: “Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: I – Theory and Software Tools”, *Atmospheric Environment*, Vol. 37, p. 5083–5096, 2003.
3. D. Daescu, A. Sandu, and G.R. Carmichael: “Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: II – Validation and Numerical Experiments”, *Atmospheric Environment*, Vol. 37, p. 5097–5114, 2003.
4. A. Sandu and R. Sander: “KPP – User’s Manual”, <http://www.cs.vt.edu/~asandu/Software/Kpp>.
5. A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, and F.A. Potra: “Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock methods”, *Atmospheric Environment*, 31:3459–3472, 1997.

6. A. R. Curtis and W. P. Sweetenham: "Facsimile/Chekmat User's Manual", Computer Science and Systems Division, Harwell Lab., Oxfordshire, Great Britain, Aug. 1987
7. R. Djouad and B. Sportisse and N. Audiffren: "Reduction of multiphase atmospheric chemistry", *Journal of Atmospheric Chemistry*, 46:131-157, 2003.
8. R. von Glasow and R. Sander and A. Bott and P. J. Crutzen: "Modeling halogen chemistry in the marine boundary layer. 1. Cloud-free MBL", *Journal of Geophysical Research*, 107(D), EID:4341, DOI:10.1029/2001JD000942, 2002.
9. R. von Kuhlmann and M. G. Lawrence and P. J. Crutzen and P. J. Rasch: "A model for studies of tropospheric ozone and nonmethane hydrocarbons: Model description and ozone results", *Journal of Geophysical Research*, 108(D), DOI:10.1029/2002JD002893, 2003.
10. R. Sander and A. Kerkweg and P. Jöckel and J. Lelieveld: "Technical Note: The new comprehensive atmospheric chemistry module MECCA", *Atmospheric Chemistry and Physics*, 5:445-450, 2005.
11. J. Trentmann and M. O. Andreae and H.-F. Graf: "Chemical processes in a young biomass-burning plume", *Journal of Geophysical Research*, 108(D), DOI:10.1029/2003JD003732, 2003.
12. Y. Tang and G. R. Carmichael and I. Uno and J.-H. Woo and G. Kurata and B. Lefer and R. E. Shetter and H. Huang and B. E. Anderson and M. A. Avery and A. D. Clarke and D. R. Blake: "Impacts of aerosols and clouds on photolysis frequencies and photochemistry during TRACE-P: 2. Three-dimensional study using a regional chemical transport model", *Journal of Geophysical Research*, 108(D), DOI:10.1029/2002JD003100, 2003.
13. E. Hairer and G. Wanner: "Solving Ordinary Differential Equations I. Nonstiff Problems", Springer Series in Computational Mathematics, 1991.
14. E. Hairer and G. Wanner. Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Springer Series in Computational Mathematics, 1996.
15. J.R. Leis and M.A. Kramer: "ODESSA - An Ordinary Differential Equation Solver with Explicit Simultaneous Sensitivity Analysis", *ACM Transactions on Mathematical Software*, Vol. 14, 1:61, 1986.
16. K. Radhakrishnan and A. Hindmarsh: "Description and use of LSODE, the Livermore solver for differential equations", Lawrence Livermore National laboratory Report UCRL-ID-113855, 1993.
17. P.N. Brown and G.D. Byrne and A.C. Hindmarsh: "VODE: A Variable Step ODE Solver", *SIAM J. Sci. Stat. Comput.*, 10:1038-1051, 1989.
18. W.P.L. Carter: "Documentation of the SAPRC-99 Chemical Mechanism for VOC Reactivity Assessment", California Air Resources Board Contract, 92-329, 2000.
19. <http://pdfcentral.shriver.umbc.edu/AutoChem/>
20. <http://www.reactiondesign.com/products/open/chemkin.html>
21. <http://www.llnl.gov/CASC/odepack/>
22. <http://mcm.leeds.ac.uk/MCM/>
23. <http://www.gnu.org/copyleft/gpl.html>
24. [http://www.cs.vt.edu/~sim\\$asandu/Software/Kpp](http://www.cs.vt.edu/~sim$asandu/Software/Kpp)
25. <http://www.mathworks.com/products/matlab/>

All-Electron DFT Modeling of SWCNT Growth Initiation by Iron Catalyst

G.L. Gutsev¹, M.D. Mochena¹, and C.W. Bauschlicher, Jr.²

¹Department of Physics, Florida A&M University, Tallahassee, Florida 32307
gennady.gutsev@famuc.edu

²Mail Stop 230-3 NASA Ames Research Center, Moffett Field, CA 94035

Abstract. Electronic and geometrical structures of $\text{Fe}_4\text{C}_n(\text{CO})_m$ ($n+m \leq 6$) and Fe_4C_n ($n=7-16$) clusters along with their singly negatively and positively charged ions are computed using density functional theory with generalized gradient approximation (DFT-GGA). Isomers with CO bonded directly to the iron atoms and bonded to a carbon atom chemisorbed on the cluster surface are optimized for the $\text{Fe}_4\text{C}_2\text{CO}$, $\text{Fe}_4\text{C}_2(\text{CO})_2$, $\text{Fe}_4\text{C}_3\text{CO}$, and $\text{Fe}_4\text{C}_4\text{CO}$ series. The computed total energies are used to estimate the energetics of the Boudouard disproportionation reactions $\text{Fe}_4\text{C}_n(\text{CO})_m + \text{CO} \rightarrow \text{Fe}_4\text{C}_{n+1}(\text{CO})_{m-1} + \text{CO}_2$. Optimizations of the Fe_4C_4 – Fe_4C_{16} clusters have shown that dimers C_2 are formed in the lowest energy states of Fe_4C_4 , trimers C_3 – in Fe_4C_5 and Fe_4C_6 , tetramers C_4 – in Fe_4C_7 and Fe_4C_8 , a pentamer C_5 – in Fe_4C_9 , and a hexamer C_6 – in Fe_4C_{10} . C_n rings attached to a Fe_3 face are formed in the lowest energy states of Fe_4C_n beginning with $n=11$.

1 Introduction

Single-walled carbon nanotubes (SWCNT) are expected to have a huge impact on various technological areas related to fabrication of sensors, composite materials, hydrogen storages, and computer memories. They can be grown using different methods typically from hydrocarbons [1], alcohol [2], and graphite [3]. The method of catalytic chemical vapor deposition [4] (CCVD) makes use high temperatures to produce carbon and a supported catalyst [5] to initiate the carbon nanotube growth. Carbon nanotubes grown using this method are often multi-walled while probably more advantageous for technology are single-walled. This process is cyclic and involves restoration of catalyst after each production cycle.

The high-pressure high-temperature HiPco [6] method is continuous and its floating or gas-phase catalyst is formed *in situ* from iron pentacarbonyl $\text{Fe}(\text{CO})_5$. Using carbon monoxide as feedstock is rather intriguing, because CO is one of the most stable diatomics with the experimental bond strength of 11.09 eV. Carbon nanotubes produced in this process are always single-walled and are believed [7] to nucleate and grow via the Boudouard disproportionation reaction $\text{CO} + \text{CO} \rightarrow \text{C}_{\text{SWCNT}} + \text{CO}_2$.

The mechanism of carbon nanotube growth is not well understood. Recent experiments [8] using epitaxial precipitation of carbon on catalyst surfaces provided evidence that open tubes drive the carbon nanotube growth. However, Jost et al. [9]

discovered that in the CCVD approach both cap and circumference growth occur. An empirical model of nucleation [10] in a CCVD process assumes that carbon saturates melted catalyst particles and a curved graphene sheet begins to form, which evolves into a SWCNT cap and subsequently lifts off the catalyst surface. The results of molecular dynamics simulations [2,11,12] of carbon nanotube growth from carbon dissolved in a catalyst particle are in agreement with this model. Other theoretical studies [13,14] dealt with the SWCNT growth from different carbon species including carbon flakes composed of hexagons and pentagons over Ni catalyst. These computations were performed using pseudopotentials and the plane-wave formalism.

2 DFT-GGA Simulations of HiPco

In order to gain insight into carbon nucleation in the HiPco process, we have performed all-electron density functional theory calculations with a generalized gradient approximation (DFT-GGA) on Fe_nC [15], Fe_nCO ($n \leq 6$) [16], and $\text{Fe}_4\text{C}_n(\text{CO})_m$ ($n+m \leq 3$, both C and CO are attached to Fe_4) [17] clusters, and estimated the energies of Boudouard CO disproportionation reactions $\text{Fe}_4\text{C}_n(\text{CO})_m + \text{CO} \rightarrow \text{Fe}_4\text{C}_{n+1}(\text{CO})_{m-1} + \text{CO}_2$. The energies found are relatively small and are in the range from -0.26 eV [$\text{Fe}_4\text{C}(\text{CO})^+$] to $+0.74$ eV [$\text{Fe}_4(\text{CO})_3$]. While most of the nanotube growth experiments to date most likely involve neutral systems, we consider cations and anions in addition to the neutrals. If the reactivity of the cation or anion appears to be much higher than that of the neutral, it would be advisable to alter the experimental conditions to enhance the number of cations or anions.

Next, we explored [18] the energetics of CO attachment to a carbon atom that had already precipitated on an iron cluster due to a preceding Boudouard reaction and evaluated the energetics of CO disproportionation reactions $\text{Fe}_4\text{C}_n(\text{CO})_m + \text{CO} \rightarrow \text{Fe}_4\text{C}_{n+1}(\text{CO})_{m-1} + \text{CO}_2$ for higher coverage ($n+m \leq 5$). We chose Fe_4^- , Fe_4 , and Fe_4^+ clusters since the Fe_4^+ cluster is known [19] to catalyze the growth of benzene from ethylene and cyclopropane in a low-pressure gas-phase process. The number of iron atoms in clusters formed in the HiPco process ranges from ~ 10 to ~ 300 and it is not clear if smaller iron clusters Fe_3 to Fe_{10} can technologically be effective for the SWCNT growth because of different restrictions such as coalescence of small clusters. Our computations show that the binding ability of clusters Fe_4 , Fe_5 , and Fe_6 is nearly the same. Therefore, one can anticipate that our computations on using computationally manageable Fe_4 clusters are capable of reproducing the essentials in initiating the SWCNT growth.

Our computations [18] arrived at several conclusions: (i). CO prefers attachment to a C atom precipitated on the Fe_4 surface in the ground state $\text{Fe}_4\text{C}_2(\text{CO})^+$ and $\text{Fe}_4\text{C}_3(\text{CO})^+$ cations while CO is attached directly to iron atoms in the neutral ground states of the $\text{Fe}_4\text{C}_2(\text{CO})$ and $\text{Fe}_4\text{C}_3(\text{CO})$ clusters; (ii). Dimerization of carbon atoms is preferable in Fe_4C_4 , Fe_4C_5 and its ions, while C_3 trimers form in the ground states of Fe_4C_6 and its ions; (iii). Formation of a C_6 ring catalyzed by Fe_4 is energetically unfavorable and it is unlikely that this ring will initiate growth of an SWCNT at least for an iron cluster of this size; (iv). Antiferromagnetic (AF) states corresponding to small spin multiplicities may closely approach ferromagnetic states with larger spin multiplicities. For example, the AF states of Fe_4C_6 with $2S+1=5$ and $2S+1=1$ are above its

ferromagnetic (FM) $2S+1=13$ ground state by +0.38 eV and +0.74 eV, respectively. This is in contrast with Fe_4 , where AF states were found at energies exceeding 1 eV.

3 Geometry Patterns in Fe_4C_n

Since attachment of COs followed by subsequent Boudouard reaction leads to formation of carbon atoms, we consider next the trends in rearrangement of carbon atoms in the lowest energy isomers of clusters Fe_4C_n , $n=7-15$. As before, computations are performed at the BPW91 level of DFT-GGA theory where B means the Becke's exchange [20] and PW91 stands for the Perdew-Wang's correlation [21] functionals. We use the 6-311+G* basis set [22]. We have optimized neutral as well as singly charged negative and positive ions of Fe_4C_n for $n=7-16$ and found that the lowest energy isomers possess the same geometrical shape independent of the charge state.

Figure 1 shows isomers of Fe_4C_7 , Fe_4C_7^- , and Fe_4C_7^+ . As is seen, the lowest energy isomers are nonplanar and contain a ring formed by C_3 , C_4 , and two iron atoms. The corresponding AF isomers have similar geometrical structures and are placed in total energy by ~ 0.5 eV higher, while an isomer of Fe_4C_7 in which C_4 is replaced by a C_3 triangle and a C atom is significantly higher. The ground-state spin multiplicities [23, 24] of bare clusters Fe_4 , Fe_4^- , Fe_4^+ are 15, 16, and 12, respectively. Attachment of two carbon chains decreases the spin multiplicity of Fe_4 , Fe_4^- , and does not change the spin multiplicity of Fe_4^+ . Similar behavior of the spin multiplicities is found for carbon attachments in Fe_4C_n for $n>7$.

The ground states of Fe_4C_8 and its ions have similar rings as in the ground states of the Fe_4C_7 series, where the C_3 ring is replaced by a C_4 ring. The spin multiplicity of Fe_4C_8 and Fe_4C_8^- are the same as in Fe_4C_7 and Fe_4C_7^- , respectively, while the ground state of the Fe_4C_8^+ cation is AF and has $2S+1=5$. This trend continues in the Fe_4C_9 series. The ground-state Fe_4C_9 neutral is presented in Fig. 2 and possesses a chair-type geometrical structure.

In order to form the ground-state Fe_4C_{10} neutral, an additional carbon atom attaches to the C_5 unit and attaches to Fe_4 from below. In Fe_4C_{11} , a C_{11} ring attached to a Fe_3 face corresponds to the lowest energy isomer. Let us note, that a C_{10} ring attached to a Fe_3 face forms an isomer of Fe_4C_{10} that is +1.86 eV higher in total energy than the isomer shown in Fig. 2. In the lowest energy isomer of Fe_4C_{16} , a carbon C_{16} ring attaches to a Fe_3 face (see Fig. 2). This trend continues in Fe_4C_{17} , and is expected to stop at C_{20} where bowl, cage, and ring isomers are found [25] to be energetically indistinguishable.

The dependence of total energies of Fe_4C_n on the spin multiplicities, or, in other words on excess spin densities at iron sites, shows no clear pattern. Thus, the difference in total energies of Fe_4C_7 when moving from the lowest $2S+1=13$ state to the lowest $2S+1=1$ state is 1.00 eV, that is, the same as in the bare iron cluster Fe_4 , while for Fe_4C_6 and Fe_4C_{12} small differences of 0.38 and 0.50 eV are found, respectively.

Figure 2 shows also the ground state of the corresponding neutral C_n species optimized at the same BPW91/6-311+G* level of theory. In agreement with experiment [26], we found that ground-state C_n and C_n^- are linear for $n=3-9$ and have ring forms of C_{2v} , C_{2h} , or C_s symmetry for $n=10-20$. In the cation series, the ring configurations are the most stable ones beginning with $n=8$.

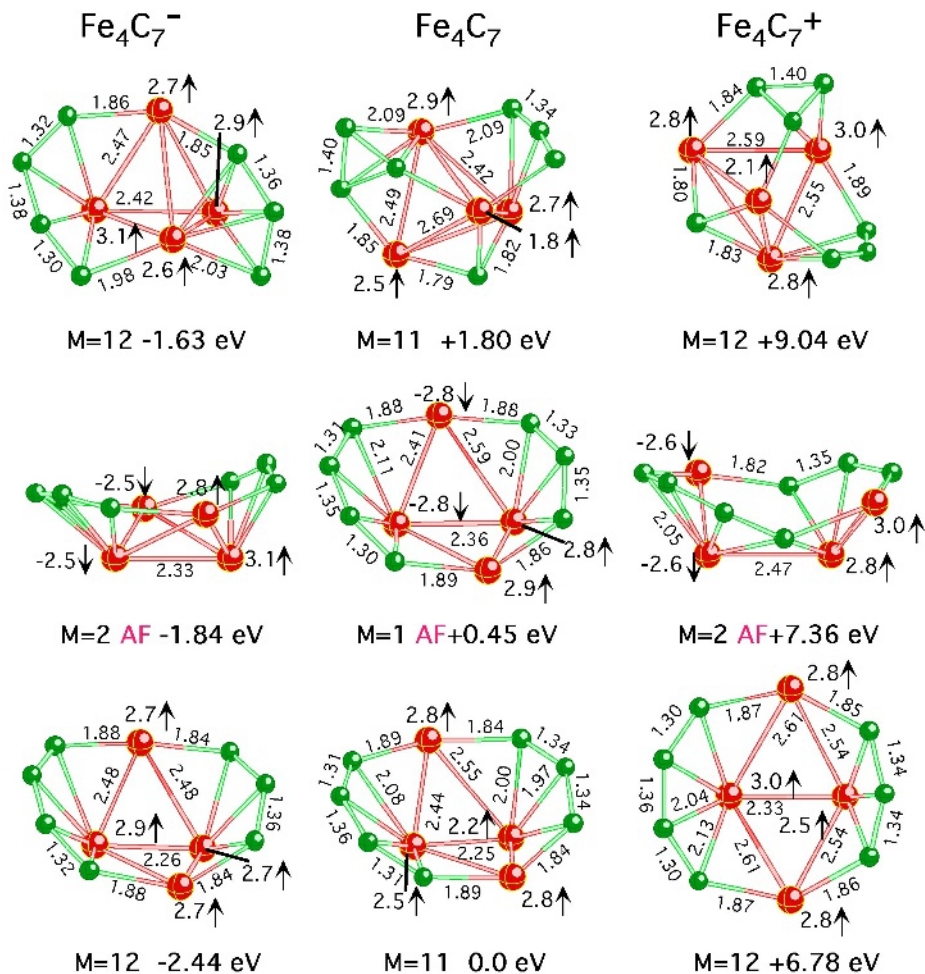


Fig. 1. Bond lengths are in Å, excess spin densities at iron atoms are in electrons, and M is the spin multiplicity $2S+1$. AF means state where excess spin densities at iron sites are coupled antiferromagnetically.

In accord with this geometrical shape change in C_n , the carbon atoms forms dimers, trimers, tetramers and pentamers in Fe_4C_n for $n=4-9$, a tetramer and an hexamer in intermediate Fe_4C_{10} , and rings beginning with $n=11$.

Since the ring structures of C_n are only preferable up to $n=20$, one expects that further carbon attachment would lead to carbon sheets, which is next step toward the growth of a carbon nanotube. The computations of Fe_4C_n for $n>16$ are currently under way.

Let us consider briefly the thermodynamic stability of Fe_4C_n and C_n taking first Fe_4C_7 as an example:

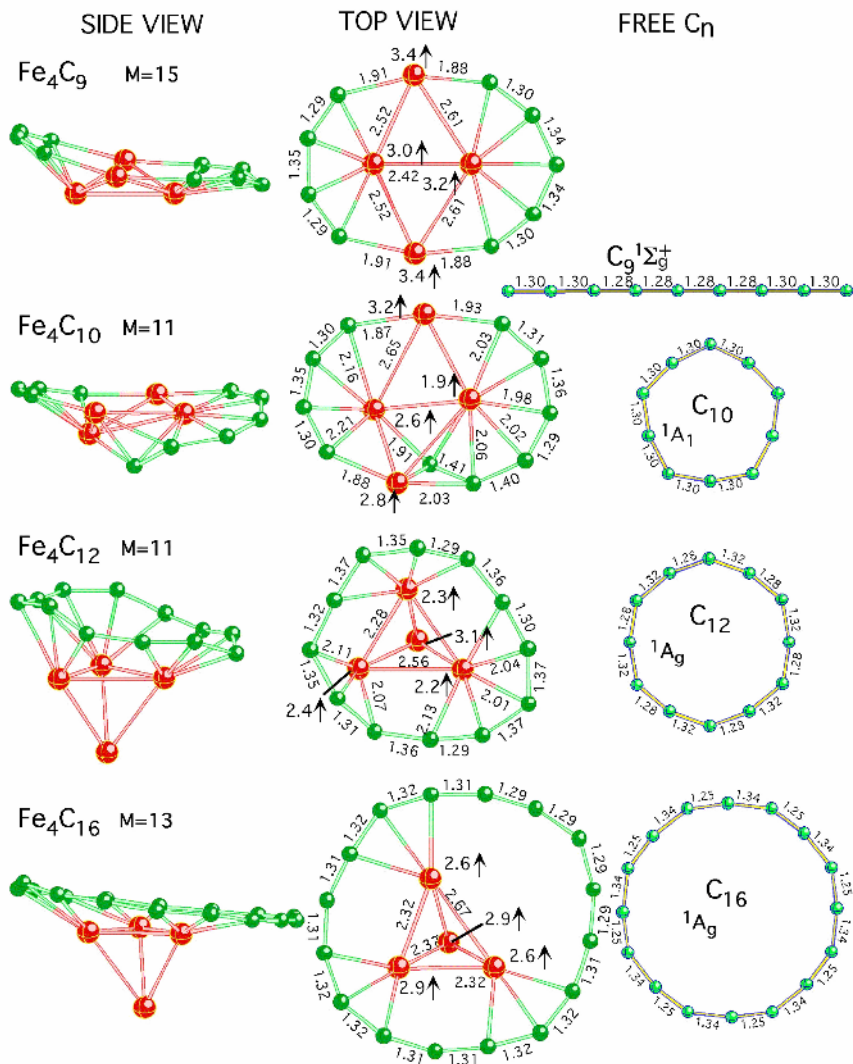
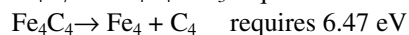
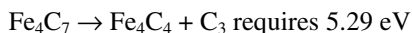
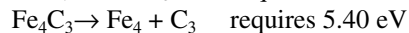
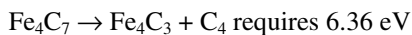


Fig. 2. Ground states of neutral Fe_4C_9 , Fe_4C_{10} , Fe_4C_{12} , Fe_4C_{16} together with ground states of C_9 , C_{10} , C_{12} , and C_{16} . Bond lengths are in Å, excess spin densities at iron atoms are in electrons, and M is the spin multiplicity $2S+1$. Note that pure C_n are 0.8 scaled in order to fit the figure.

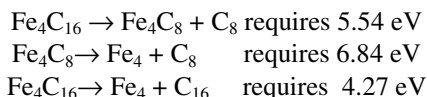


Complimentary channels show a similar trend of a larger binding energy for C_4 than C_3 .



The channel $C_3 + C_4 \rightarrow C_7$ produces 6.18 eV, which results in the energy of the $Fe_4C_7 \rightarrow Fe_4 + C_7$ channel of 5.58 eV. Of course, the same energy is obtained in the direct computation $E_{tot}(Fe_4C_7) - E_{tot}(Fe_4) - E_{tot}(C_7)$. One sees that the binding energies of C_n to Fe_4 or Fe_4C_m ($n, m = 3$ or 4) are rather similar to the energy gained from $C_3 + C_4 \rightarrow C_7$.

Beginning with $n=11$, the carbons form a ring and the additional C—C bonds may influence thermodynamic patterns. Let us consider, as an example, the decay channels of Fe_4C_{16} .



That is, the detachment of C_8 from the ring in Fe_4C_{16} is less costly than the detachment of C_8 from the iron cluster in Fe_4C_8 and the C_{16} ring is less bound to Fe_4 than are the two C_4 chains in the Fe_4C_8 cluster. The ring formation becomes favorable due to the high energy of dimerization $C_8 + C_8 \rightarrow C_{16}$, which is 8.11 eV according to our computations at the same level of theory, and not from a stronger bonding of the ring to the cluster.

4 Conclusion

The results of our computations using density functional theory on Fe_4C_n , $n=7-16$, revealed that

- (i) The carbon atoms forms two chains attached in such a way as to form a ring containing two iron atoms for $n=7, 8$ and 9 .
- (ii) Carbon atoms form rings on the top of a Fe_3 face for $n=11-16$.
- (iii) Fe_4C_{10} presents an intermediate state between these (i) and (ii) where C_4 and C_6 do not form a ring.
- (iv) The trend in preferable deposition of carbon atoms on the Fe_4 cluster is similar to the trend in geometrical shape of free C_n species, where the linear—ring transition occurs at $n=10$.
- (v) The iron catalyst in the HiPco process reduces significantly the barrier for the Boudouard reaction which releases carbon atoms followed by their reconstruction into rings at the catalyst surface. After the ring template is created on the catalyst, it is likely that all subsequent carbon atom production on the catalyst leads to growth of the SWNT. However, there could well be a tradeoff between growth rate and the number of defects.

Acknowledgement

This work was supported in part by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under cooperative Agreement DAAD19-01-2-0014. The content of which does not necessary reflect the position or the policy of the government, and no official endorsement should be inferred. We thank Dr. Norbert Müller

for providing us with the new version of Ball&Stick software [N. Müller, Ball&Stick 4.0 (pre-release), molecular graphics software for MacOS X, Johannes Kepler University Linz, 2004.<http://www.orc.uni-linz.ac.at/mueller/ball_and_stick>] used for plotting individual geometrical structures presented in the figures.

References

1. Su, M., Liu, J.: A scalable CVD method for the synthesis of single-walled carbon nanotubes with high catalyst productivity, *Chem. Phys. Lett.* 322 (2000) 321-326.
2. Maruyama, S., Murakami, Y., Shibuta, Y., Miyauchi, Y., Chiashi, S.: Generation of Single-Walled Carbon Nanotubes from Alcohol and Generation Mechanism by Molecular Dynamics Simulations, *J. Nanosci. Nanotech.* 4 (2004) 360-367.
3. Arepalli, S.: Laser Ablation Process for Single-Walled Carbon Nanotube Production, *J. Nanosci. Nanotech.* 4 (2004). 317-325.
4. Dai, H., Rinzler, A.G., Nikolaev, P., Thess, A., Golbert, D.T., Smalley, R.E.: Single-wall nanotubes produced by metal-catalyzed disproportionation of carbon monoxide, *Chem. Phys. Lett.* 260 (1996) 471-475.
5. Kong, J., Cassel, A.M., Dai, H.: Chemical vapor deposition of methane for single-walled carbon nanotubes. *Chem. Phys. Lett.* 292 (1998) 567-574.
6. Nikolaev, P., Bronikowski, M.J., Bradley, R.K., Rohmund, F., Colbert, D.T., Smith, K.A., Smalley, R.E.: Gas-phase catalytic growth of single-walled carbon nanotubes from carbon monoxide, *Chem. Phys. Lett.* 313 (1999) 91-97.
7. Nikolaev, P.: Gas-Phase Production of Single-Walled Carbon Nanotubes from Carbon Monoxide: A Review of the HiPco Process, *J. Nanosci. Nanotech.* 4 (2004) 307-316.
8. Tsui, F., Ryan, P.A.: Self-Organization of Carbide Superlattice and Nucleation of Carbon Nanotubes, *J. Nanosci. Nanotech.* 4 (2004) 408-413.
9. Jost, O., Gorbunov, A., Liu, X., Pompe, W., Fink, J.: Single-Walled Carbon Nanotube Diameter, *J. Nanosci. Nanotech.* 4 (2004) 433-440.
10. Gavillet, J., Thibault, J., Stéphan, O., Amara, H., Loiseau, A., Bichara, C., Gaspard, J.-P., Ducastelle, F.: Nucleation and Growth of Single-Walled Nanotubes: The Role of Metallic Catalysts, *J. Nanosci. Nanotech.* 4 (2004) 346-359.
11. Zhao, J., Martinez-Limia, A., Balbuena, P. B. Understanding Catalysed Growth of Single-Wall Carbon Nanotubes, *Nanotechnology*, 16, (2005). S575-S581.
12. Ding, F., Rosén, A., Bolton, K.: Molecular dynamics study of the catalyst particle size dependence on carbon nanotube growth, *J. Chem. Phys.* 121 (2004) 2775-2779.
13. Fan, X., Buczko, R., Puzos, A.A., Geohegan, D.B., Howe, J.Y., Pantelides, S.T., Pennycook, S.J.: Nucleation of Single-Walled Carbon Nanotubes, *Phys. Rev. Lett.* 90 (2003) 145501-145504.
14. Wells, J.C., Noid, D.W., Sumpter, B.G., Wood, R.F., Zhang, Q.: Multiscale simulations of carbon nanotube nucleation and growth: Electronic structure calculations: *J. Nanosci. Nanotech.* 4 (2004) 414-422.
15. Gutsev, G.L., Bauschlicher, Jr., C.W. : Structure of neutral and charged Fe_nCO clusters ($n = 1-6$) and energetics of the $\text{Fe}_n\text{CO} + \text{CO} \rightarrow \text{Fe}_n\text{C} + \text{CO}_2$ reaction, *J. Chem. Phys.* 119 (2003) 368-3690.
16. Gutsev, G.L., Bauschlicher, C.W., Jr.: Interaction of carbon atoms with Fe_n , Fe_n^- , and Fe_n^+ clusters ($n=1-6$), *Chem. Phys.* 291 (2003) 27-40.
17. Gutsev, G.L., Mochena, M.D., Bauschlicher, C.W., Jr.: Structure and properties of Fe_4 with different coverage by C and CO, *J. Phys. Chem.* 108 (2004) 11409-11418.

18. Gutsev, G.L., Mochena, M.D., Bauschlicher, C.W., Jr.: All-electron DFT modeling of SWCNT growth on iron catalysts from carbon monoxide feedstock, *J. Nanosci. Nanotech.*, in press.
19. Schnabel, P., Irion, M.P., Weil, K.G.: Evidence for low-pressure catalysis in the gas phase by a naked metal cluster: the growth of benzene precursors on iron (Fe_4^+), *J. Phys. Chem.* 95 (1991) 9688-9694.
20. Becke, A.D.: Density-functional exchange-energy approximation with correct asymptotic behavior, *Phys. Rev. A* 38 (1988) 3098-3100.
21. Perdew, J.P., Wang, Y.: Accurate and simple analytic representation of the electron-gas correlation energy, *Phys. Rev. B* 45 (1992) 13244-13249.
22. Raghavachari, K., Trucks, G.W.: Highly correlated systems: Excitation energies of first row transition metals Sc–Cu, *J. Chem. Phys.* 91 (1989) 1062-1065.
23. Chrétien, S., Salahub, D.R.: Kohn-Sham density-functional study of low-lying states of the iron clusters $\text{Fe}_n^+/\text{Fe}_n/\text{Fe}_n^-$, *Phys. Rev. B* 66 (2002) 155425-155437.
24. Gutsev, G.L., Bauschlicher, C.W., Jr.: Electron affinities, ionization energies, and fragmentation energies of Fe_n clusters ($n = 2-6$): A Density Functional Theory study, *J. Phys. Chem. A* 107 (2003) 7013-7023.
25. An, W., Bulusu, S., Zeng, X.C.: Ab initio calculation of bowl, cage, and ring isomers of C_{20} and C_{20}^- , *J. Chem. Phys.* 204109 (2005) 1-8.
26. Yang, S., Taylor, K.J., Craycraft, M.J., Conceicao, J., Pettiette, C.L., Cheshnovsky, O., Smalley, R.E.: UPS of 2–30-atom carbon clusters: Chains and rings, *Chem. Phys. Lett.* 144 (1988) 431-436.

Ab initio Study of Chiral Recognition of β -Butyrolactone by Cyclodextrins

Waraporn Parasuk^{1,*} and Vudhichai Parasuk²

¹ Department of Chemistry, Faculty of Science,
Kasetsart University, Bangkok 10900, Thailand
fsciwapa@ku.ac.th

² Department of Chemistry, Faculty of Science,
Chulalongkorn University, Bangkok 10330, Thailand
parasuk@atc.atccu.chula.ac.th

Abstract. Separation of stereoisomers of organic compounds is an important and challenge task for chemists. Cyclodextrins and their derivatives have been widely used in chromatography for this application. Experimental results indicated that substituents on the hydroxyl groups of cyclodextrin affect the efficiency of the chiral separation of β -butyrolactone. The understanding of the interactions contributed to the chiral recognition of cyclodextrin would help us predict the separation capability of a specific pair of cyclodextrin and chiral compound. Thus, the cyclodextrin substituent effect on the chiral recognition should be systematically investigated. In this study, Hartree Fock method with 3-21G basis set and density functional theory B3LYP with 6-31G* basis set were applied to determine the chiral recognition of a chiral model, β -butyrolactone, by β -cyclodextrin and its derivatives. Both methods predicted comparable values of chiral recognition of β -cyclodextrin derivatives. We found that methoxyl substitution on the wider rim of cyclodextrin (secondary hydroxyl groups) give the most effective chiral separation ($\Delta\Delta E=18.2$ kcal/mol in favor of R-isomer) followed by substitution on the narrow rim ($\Delta\Delta E=9.5$ kcal/mol in favor of S-isomer) while substitution on both side give the worst recognition ($\Delta\Delta E=3.2$ kcal/mol in favor of S-isomer). This suggests that β -cyclodextrin with substitution only on the wider rim give the best chiral selectivity. By replacing methyl group with chiral hydroxypropyl group, we found that the chiral selectivity is reduced ($\Delta\Delta E=6.4$ and 8.4 kcal/mol respectively for R- and S-form of hydroxypropyl group). This implies that the bulky group causes the reduction of the chiral selectivity.

1 Introduction

Cyclodextrins (CDs) are cyclic α -1,4 linked oligosaccharides of D-glucose units. The most naturally available are α -, β -, and γ -CDs with 6, 7, and 8 glucose units. Such a linkage forms a truncated cone shape of CDs with the 6-primary hydroxyl groups of glucose at the narrower and the 2- and 3- secondary hydroxyl groups at the wider rim. The cavity of CDs is less polar than other parts of molecule.[1] The cavity size varies from 5 to 9 Å depending on numbers of glucose units. CDs can form inclusion

complexes with several classes of substances. Several factors govern the stability of the complexes. Since CDs have a lot of chiral centers, they can form diastereomers with enantiomeric guests with different stability. Thus, CDs are utilized as reagents for chiral separation. One of such applications is the usage of CDs as either stationary or mobile phase in chromatographic separation of isomers.[2] Several classes of enantiomeric compounds i.e. sugars, alcohols, amines, lactones, epoxides, bicyclic compounds, were gas chromatographic resolved on derivatized CDs column.[3,4] This capability of cyclodextrin in chromatography has been widely applied to pharmaceutical and food additive industries, fragrances, and pheromone research.[2,5] Shitangkoon and Vigh reported that trichloroacetyl substituents on β -cyclodextrin, as stationary phase in gas chromatography, gives good column efficiency while monochloroacetyl yields good selectivity of the chiral separation of β -butyrolactone.[6] They found furthermore that dichloroacetyl substitution gives the compromised results of the column efficiency and the chiral selectivity. Substituents seem to have an effect on the chiral separation of the chromatographic column. To understand the substitution effect on the chiral separation by cyclodextrin, the inclusion phenomenon between β -cyclodextrins and β -butyrolactone were investigated using computational modeling techniques.

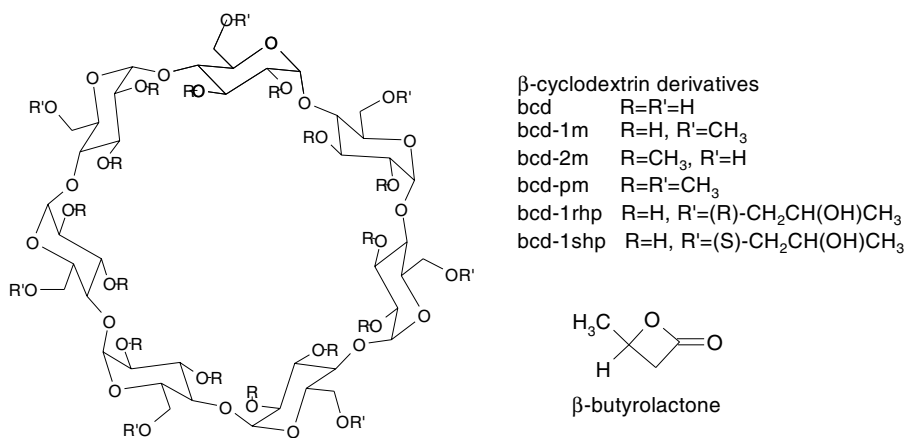


Fig. 1. Structure of β -cyclodextrin derivatives and β -butyrolactone

2 Computational Details

Illustrative representation of β -cyclodextrin derivatives and β -butyrolactone are displayed in Figure 1. Cyclodextrins studied in this work were β -cyclodextrin, denoted as bcd, and its derivatives. The derivatives were modified by substitution of hydroxyl with methoxyl groups. The suffixes 1m, 2m, or pm expressed for methoxyl substituted at primary, secondary, or both rims of the parent cyclodextrin, i.e. bcd-1m, bcd-2m, and bcd-pm stand for per(6-O-methyl)- β -cyclodextrin, per(2,3-di-O-methyl)- β -cyclodextrin, and per(2,3,6-tri-O-methyl)- β -cyclodextrin, respectively. Moreover, the substitution of

the primary hydroxyl with chiral substituents, R- and S-2-hydroxypropoxyl, i.e. per(6-O-R-2-hydroxymethyl)- β -cyclodextrin and per(6-O-S-2-hydroxymethyl)- β -cyclodextrin denoted as bcd-1rhp and bcd-1shp, were also studied. Geometries of host β -cyclodextrin and derivatives, together with those of enantiomers of guest β -butyrolactone, and their host-guest complexes were taken from our previous work where molecular dynamics simulations of the inclusion complexes between bcd and β -butyrolactone were performed.[7] In that study, two orientations of β -butyrolactone i.e. one with methyl group pointing inward and another with methyl group pointing outward the secondary end of the cyclodextrin, were considered. In this work, energies of the more stable complexes, hosts, and guests were recalculated at Hartree Fock level with 3-21g basis set and density functional theory level using B3LYP functionals with 6-31g* basis set. The *ab initio* and DFT calculations were performed using Gaussian98 program [8].

3 Results and Discussion

3.1 Binding Energies

Binding energies, ΔE , of the inclusion complexes were computed from the energy differences between complexes and corresponding host and guest molecules.

$$\Delta E_{R/S} = E_{R/Scpx} - (E_{host} + E_{R/S})$$

$E_{R/Scpx}$ is the energy of R- or S- β -butyrolactone and cyclodextrin complex. E_{host} is the energy of free cyclodextrin derivative, and $E_{R/S}$ is the energy of guest R- or S- β -butyrolactone. Table 1 displays binding energies of inclusion complexes between bcd derivatives and R- and S- β -butyrolactone. From the table, HF and B3LYP gave similar trend of binding energies, even though HF seems to overestimate the binding energies as compared to the results obtained at B3LYP level. For instances, HF predicted both bcd and bcd-2m bind to S- β -butyrolactone with the binding energy of -5.3 and -2.7 kcal/mol, respectively, while B3LYP predicted that S- β -butyrolactone does not bind with bcd and bcd-2m ($\Delta E_S = 0.4$ and 4.6 kcal/mol). The deficiency of HF/3-21g in the study of cyclodextrin complex where weak interaction is involved is clearly seen. Considering the inclusion behavior of R- β -butyrolactone, it is observed that the R-enantiomer could not bind with bcd and bcd-1m, as indicated by

Table 1. Binding energies (ΔE_R and ΔE_S , kcal/mol) of the complexes between R- and S- β -butyrolactone and β -cyclodextrin and derivatives calculated by HF/3-21g and B3LYP/6-31g*

	HF/3-21g		B3LYP/6-31g*	
	ΔE_R	ΔE_S	ΔE_R	ΔE_S
Bcd	1.9	-5.3	6.2	0.4
bcd-1m	0.5	-11.1	1.9	-7.7
bcd-2m	-22.1	-2.7	-13.6	4.6
bcd-pm	-19.4	-24.8	-14.3	-17.4

the positive ΔE_R while it forms complexes with bcd-2m and bcd-pm ($\Delta E_R = -13.6$ and -14.3 kcal/mol, respectively). On the other hand, the *S*- β -butyrolactone forms complexes with bcd-1m and bcd-pm ($\Delta E_S = -7.7$ and -17.4 kcal/mol, respectively) while it does not bind with bcd and bcd-2m. It could be seen that the complexation pattern with bcd and bcd derivatives of the *R*- and *S*-enantiomers of β -butyrolactone is distinctive. Thus, the degree of methylation of β -cyclodextrin affected the stability of the inclusion complex of β -butyrolactone differently for the different enantiomers.

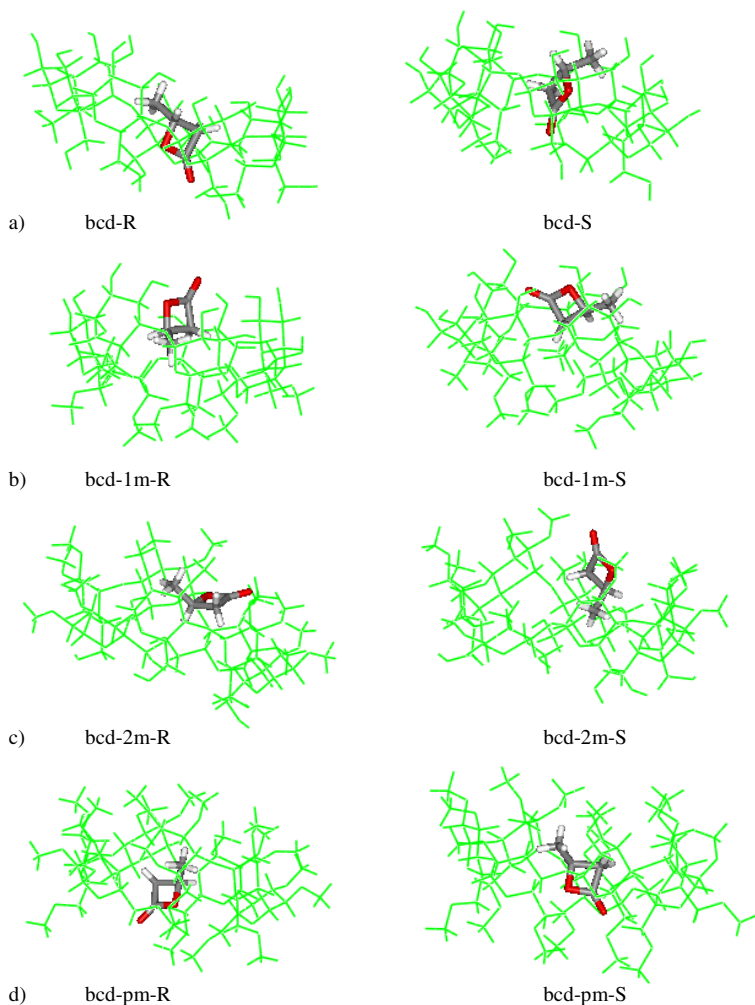


Fig. 2. Side view of inclusion complexes of *R*- and *S*- β -butyrolactone with derivatives of β -cyclodextrin, a) bcd, b) bcd-1m, c) bcd-2m, and d) bcd-pm. The cyclodextrin skeletons are presented as thin line, β -butyrolactone are presented as color stick with ● ● ○ for C, O, and H, respectively. In all figure the upper side is the wider secondary rim[7].

Figure 2 shows structures of inclusion complexes of bcd and its derivatives with R- and S- β -butyrolactone obtained from MD simulation.[7] In the complex between bcd and R- and S- β -butyrolactone, the guests were placed inside the cavity of bcd as shown in Figure 2a. However, these are not the stable complexes since their binding energies are positive. From Figure 2b, the methyl group of the guest S- β -butyrolactone pointed outward to the wider secondary hydroxyl groups of the bcd-1m, while the methyl group of R-enantiomer pointed inward to the ring. The former forms the more stable complex than the latter. This evidence was also observed in the complexes of host bcd-2m. In figure 2c, the R-enantiomer forms the more stable complex ($\Delta E_R = -13.6$ kcal/mol) and its methyl group points outward to the wider end of the cyclodextrin ring. The S-enantiomer which forms less stable complex, however, has its methyl group points inward to the cyclodextrin ring. The host bcd-pm, in which the hydroxyl groups on both ends are methylated, forms complexes with both enantiomers with comparable stability. As displayed in figured 2d, the orientation of the R- and S-guests in both complexes are similar, the guest situated inside the cavity and the methyl group of the guest points outward to the secondary end. It is interesting to note that stable complexes of bcd-1m (with S-enantiomer) and bcd-2m (with R-enantiomer) have the methyl group point outwad the bcd ring while the unstable ones have the methyl group points inward. This is also the case for bcd-pm where both R- and S-enantiomers have the methyl group point outward the secondary methoxyl rim. The orientation of the guest is very important for the formation of the stable complex.

3.2 Chiral Recognitions

The chiral recognition of cyclodextrin ($\Delta\Delta E$), exhibited in Table 2, was computed from the difference of the binding energies of both enantiomers, ΔE_R and ΔE_S , or from the energy difference of their inclusion complex with R- and S- β -butyrolactone,

$$\Delta\Delta E = \Delta E_R - \Delta E_S \quad \text{or} \quad \Delta\Delta E = E_{R_{\text{cpX}}} - E_{S_{\text{cpX}}}$$

where $E_{R_{\text{cpX}}}$ and $E_{S_{\text{cpX}}}$ are energies of corresponding R- and S- β -butyrolactone cyclodextrin complexes. Larger $\Delta\Delta E$ means more efficient chiral separation. The bcd-2m showed the best chiral recognition with $\Delta\Delta E$ of -19 and -18 kcal/mol by HF and B3LYP methods, respectively. Though the different level of accuracy predicts different values of binding energy of the complexes, there is not much different in the chiral recognition prediction. The negative value expressed larger binding energy for R- than for S- configuration and vice versa for the positive value. Having the smaller $\Delta\Delta E$ (in magnitude), bcd-1m has a moderate value of $\Delta\Delta E$ and also could be used for chiral separation. Though a moderate value was obtained, bcd might not be suitable to use as chiral selective reagent as it gave positive value of the binding energy to both enantiomers of β -butyrolactone. Thus, the methylation of the secondary as well as the primary hydroxyl groups of cyclodextrin could improve its chiral selectivity. The methylation of the secondary hydroxyl improves the binding with R-enantiomer

whereas the methylation of the primary hydroxyl improves the binding with S-enantiomer. There are not much difference on the $\Delta\Delta E$ of bcd-1m, bcd-1rhp, and bcd-1shp (all have the substituents on the primary hydroxyl groups), i.e. 9.5, 6.4, and 8.4 kcal/mol, respectively. Positive values of $\Delta\Delta E$ confirmed that the substituents on the primary rim improve the stability of the complex of the S-enantiomer. Thus, the chiral R- or S-2-hydroxypropyl did not change the chiral preference of cyclodextrin. When the hydroxyl groups on both rims of bcd are methylated as in bcd-pm, R- and S- β -butyrolactone bind to the host favorably. They form complex with the host with comparable stability and hence leads to small $\Delta\Delta E$, not suitable to use as chiral selective reagent. Thus, the host which forms the more stable complex (greater binding energy) is the more chiral selective (in exception of bcd-pm in which it forms stable complex with both R- and S-forms). A similar conclusion could also be observed from the gas chromatographic experiment.[6] In that study, the monochloroacetyl derived bcd gives the best chiral selectivity as compared to the dichloro- and the trichloroacetyl substituted bcds since it forms the most stable complex as implying from its largest retention time.

Table 2. Chiral recognition energies ($\Delta\Delta E$, kcal/mol) of β -cyclodextrin and derivatives calculated by HF/3-21g and B3LYP/6-31g*

	$\Delta\Delta E$	
	HF/3-21g	B3LYP/6-31g*
bcd	7.1	5.8
bcd-1m	11.5	9.5
bcd-2m	-19.5	-18.2
bcd-pm	5.4	3.2
bcd-2rhp	9.5	6.4
bcd-2shp	13.4	8.4

4 Conclusions

The stability of the inclusion complexes of β -cyclodextrin derivatives with β -butyrolactone depends on the degree of methylation of β -cyclodextrin. The derivative bcd-pm, where all the hydroxyl groups are methylated, forms stable complexes with both configurations of β -butyrolactone and is not very good in the chiral recognition. From this study, bcd-2m exhibited the best capability of the chiral recognition for β -butyrolactone and is recommended to use as the chiral selective agent. The bcd, on the other hand, has positive binding energies for both R- and S- β -butyrolactone is expected to give poor chiral recognition. Though HF method seems to overestimate the binding energies of the inclusion complexes, both HF and B3LYP methods predicted comparable values of chiral recognitions of β -cyclodextrin derivatives.

Acknowledgement

The Computational Chemistry Unit Cell (CCUC), Department of Chemistry, Faculty of Science, Chulalongkorn University is acknowledged for the computational facilities.

References

1. Li, S., Purdy, W.C.: Cyclodextrins and Their Applications in Analytical Chemistry. *Chem. Rev.* 92 (1992) 1457-1470
2. Lipkowitz, K. B., Pearl, G., Coner, B., Peterson, M. A.: Explanation of Where and How Enantioselective Binding Takes Place on Permethyated β -Cyclodextrin, a Chiral Stationary Phase used in Gas Chromatography. *J. Am. Chem. Soc.* 119 (1997) 600-610
3. Armstrong, D.W., Li, W., Chang, C.-D., Pitha, J.: Polar-liquid, Derivatized Cyclodextrin Stationary Phases for the Capillary Gas Chromatography Separation of Enantiomers. *Anal. Chem.* 62 (1990) 914-923
4. Schurig, V., Nowotny, H.-P.: Gas Chromatographic Separation of Enantiomers on Cyclodextrin Derivatives. *Angew. Chem. Int. Ed. Engl.* 29 (1990) 939-957
5. Armstrong, D. W., Ward, T.J., Armstrong, R. D., Beesley, T. E.: Separation of Drug Stereoisomers by the Formation of β -Cyclodextrin Inclusion Complexes. *Science* 232 (1986) 1132-1135
6. Shitangkoon, A., Vigh, G.: Gas Chromatographic Enantiomer Separations using Chloroacylpentyl Cyclodextrins. Abstracts of the 24th Congress on Science and Technology of Thailand, Bangkok (1998) 218-219
7. Parasuk, W., Longwan, N., Tasanakosol W.: Enantiomer Recognition of β -Butyrolactone by Cyclodextrin. The fifth Annual National Symposium on Computational Science and Engineering, Bangkok (2001) 187-193
8. Gaussian 98, Revision A9, Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Zakrzewski, V. G., Montgomery, Jr., J. A., Stratmann, R. E., Burant, J. C., Dapprich, S., Millam, J. M., Daniels, A. D., Kudin, K. N., Strain, M. C., Farkas, O., Tomasi, J., Barone, V., Cossi, M., Cammi, R., Mennucci, B., Pomelli, C., Adamo, C., Clifford, S., Ochterski, J., Petersson, G. A., Ayala, P. Y., Cui, Q., Morokuma, K., Malick, D. K., Rabuck, A. D., Raghavachari, K., Foresman, J. B., Cioslowski, J., Ortiz, J. V., Baboul, A. G., Stefanov, B. B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Gomperts, R., Martin, R. L., Fox, D. J., Keith, T., Al-Laham, M. A., Peng, C. Y., Nanayakkara, A., Challacombe, M., Gill, P. M. W., Johnson, B., Chen, W., Wong, M. W., Andres, J. L., Gonzalez, C., Head-Gordon, M., Replogle, E. S., and Pople, J. A., Gaussian, Inc., Pittsburgh PA (1998)

C-H Functionalisation Through Singlet Chlorocarbenes Insertions – MP2 and DFT Investigations

M. Ramalingam¹, K. Ramasami², P. Venuvanalingam³, and V. Sethuraman¹

¹Rajah Serfoji Government College, Thanjavur-613005, India

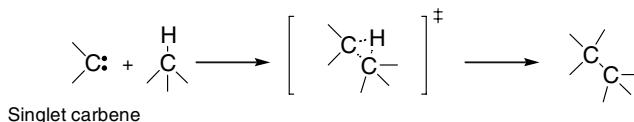
²Nehru Memorial College, Puthanampatti-621007, India

³Bharathidasan University, Tiruchirappalli-620024, India

Abstract. The insertion reactions of singlet mono and dichlorocarbenes (¹CHCl and ¹CCl₂) into primary, secondary and tertiary C-H bonds of methane, ethane, propane, *n*-butane and *iso*-butane have been investigated at ab initio (MP2) and DFT levels invoking 6-31g(d, p) basis set. Among the σ and π insertions into the said alkane C-H, both MP2 and DFT predict that the σ approach is slightly favoured over the π approach. For inverted σ approach the barrier heights have been found to be *ca.* 2 to 5 times that of the normal σ approach.

1 Introduction

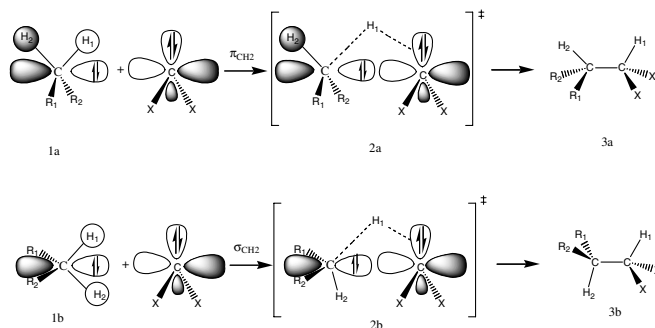
The refrigerants hydrochlorofluorocarbons (HCFCs) and hydrofluorocarbons (HFCs) are not harmful to ozone layer as they are oxidized / photodissociated to halocarbenes [1]. The halocarbenes and carbenes are important transient intermediates in organic synthesis [2], in organometallic chemistry [3] and in gas phase combustion [4, 5]. In the ground state the singlet halocarbenes are stabilized over the triplet carbenes by a large $\sigma - p_{\pi}$ separation [6, 7, 8] and inductive/ mesomeric effects or a combination of both the effects. The vacant p_{π} and the occupied σ orbitals make them electrophiles and nucleophiles respectively. The singlet carbenes follow the concerted one-step process through a three-center cyclic transition state [9] (Scheme 1). As the short lived transient carbenes can not be analysed easily by experimental methods,



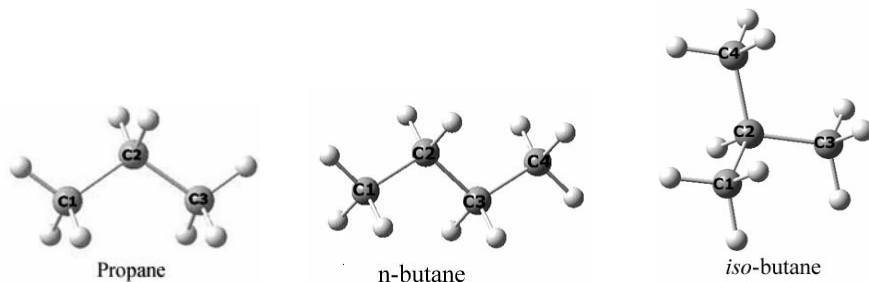
Scheme 1. Insertion pathway for singlet carbene

ab initio quantum mechanical method could be used as a powerful tool to investigate them. This is obvious from the fact that the activation barriers of 0 and 0.2 kcal/mol for ¹CH₂ insertion into the C-H of methane and ethane respectively at MP3/6-31g (d) [10-12] have been reported to be in excellent agreement with the experimental data [2,13]. But calculations at HF level overestimate the activation barrier heights due to

lack of electron correlation [14-16]. In the halocarbenes, the halogens would interact with the carbenic carbon through the oppositely operating electronic [mesomeric (+M) - π donor and inductive (-I) - σ acceptor] effects. As a result of this the electrophilicity of the halocarbenes decreases with increased halogenation [8,17]. Consequently the activation barrier increases because the primary step in the carbene insertion is an electrophilic attack of the carbene center by the C-H bond. This explains why the highly stable nucleophilic carbenes of Arduengo type [19] are completely unreactive towards the C-H bonds. Hence nucleophilic carbene insertions are generally not observed at low temperature matrixes. However certain intramolecular insertion of carbene into C-H bond at 11 K has been observed [18]. Considering the chlorocarbenes (CX_2 , $X=H, Cl$) as electrophiles, two different pathways for the C-H insertion process have been proposed according to FMO treatment [20, 21] (Scheme 2). In the π -approach the vacant p_π orbital of chlorocarbene overlaps with the filled π_{CH_2} fragment orbital of alkane (1a). In the σ -approach the vacant p_π orbital of chlorocarbene overlaps with the σ_{CH_2} fragment orbital of the alkane (1b). The activation barrier of the reaction pathway is studied taking into account the orientation of the chlorocarbene and the electronic and steric effects. With due consideration of the electronic and steric factors, the present investigation focuses on the mechanism of 1CHCl and 1CCl_2 insertion into the primary, secondary and the tertiary C-H bonds of methane, ethane, propane, n-butane and *iso*-butane (Scheme 3), as investigations of these have not been well documented.



Scheme 2. The pCH_2 and sCH_2 orientations for the insertion of singlet carbenes



Scheme 3. Compounds and numbering system adopted in this study

2 Method of Calculations

This investigation was performed with Gaussian03W suite of program [22] implemented on a Pentium IV/3.2 GHz processor. The geometries of the chlorocarbenes, the alkane substrates, the transition states and the products have been optimized initially at HF/6-31g (d, p) level. The resultant geometries were taken as inputs for MP2 and B3LYP [23-27], investigations. For a better treatment of 1, 2-hydrogen shift during the insertion process, standard 6-31g (d, p) [28, 29] basis set has been adopted. In all the π -approaches the carbenic-carbon and the atoms of the -CHR- unit of the alkane bearing the hydrogen that undergoes shift to the nucleophilic face of the carbene were constrained to be in one plane during the geometry optimization. The harmonic vibrational frequency calculations at MP2 and B3LYP levels were carried out to characterize all the stationary points as either minima, first-order transition states or second-order saddle points – SOSP.

3 Results and Discussion

3.1 Singlet Chlorocarbenes Insertion into Methane and Ethane

The activation energies analysed in this investigation have been computed at B3LYP/6-31g (d, p) and MP2/6-31g (d, p) levels and all these have been listed in column 2 to 5 of Table 1. The p_{π} of $^1\text{CHCl}$ and $^1\text{CCl}_2$ may interact with alkane's filled fragment orbital of either σ or π symmetry (Scheme 2). So chlorocarbenes insertion reactions have been investigated for both (σ/π) the approaches and the σ approach has been adjudicated to be the minimum energy path resulting in a staggered

Table 1. Activation barriers (in kcal/mol) for insertion of $^1\text{CHCl}$ and $^1\text{CCl}_2$ into alkanes at B3LYP/6-31G** and MP2/6-31G** levels for sigma (π) orientations

	CHCl		CCl ₂	
	B3LYP	MP2	B3LYP	MP2
methane	4.87(5.15)	7.05(7.35)	20.57(21.44)	18.83(19.83)
methane-sii (methane-pii)	23.59(19.58)	22.87(20.12)	43.65(39.91)	38.28(38.28)
ethane	1.27(1.82)	3.47(3.67)	15.66(16.97)	12.98(14.30)
ethane-sii (ethane-pii)	18.90(17.14)	15.14(16.82)	36.26(36.26)	33.66(33.68)
propane (C1)	1.37(1.80)	3.22(3.35)	15.90(17.38)	12.60(14.17)
propane (C2)	-0.64(0.50)	0.79(1.39)	12.45(13.80)	8.59(10.24)
n-butane(C1)	1.21(2.29)	3.05(3.44)	15.48(17.21)	11.94(13.91)
n-butane(C2)	-0.68(0.73)	0.76(1.05)	12.62(14.80)	7.95(10.30)
iso-butane (C1)	1.35(1.81)	2.62(3.01)	16.51(17.45)	11.83(13.59)
iso-butane (C2)	-0.14(0.24)	-0.73	9.40(11.52)	4.42(6.62)

sii – sigma inverted; pii – pi inverted.

conformer. Our recent investigation of fluorocarbenes insertion [17] and the earlier report of carbene [30] and oxygen insertions [31] into C-H, also preferably assume σ orientation over the π approach. The σ approaches of $^1\text{CHCl}$ and $^1\text{CCl}_2$ towards the C-H of methane are associated with the activation energies of 4.87 and 20.57 kcal/mol respectively at B3LYP/6-31g (d, p) (Table 1). The MP2/6-31g (d, p) value for $^1\text{CHCl}$ insertion is *ca.* 2 kcal/mol higher and that for $^1\text{CCl}_2$ insertion is *ca.* 2 kcal/mol lower than those of the corresponding B3LYP/6-31g (d, p) values. Replacement of hydrogen by chlorine in $^1\text{CHCl}$ decreases its electrophilicity [32]. So the barrier heights increase dramatically from 4.87 to 20.57 kcal/mol at B3LYP and 7.05 to 18.83 kcal/mol at MP2 calculations respectively for methane. Both in the σ and the π orientations the enhanced nucleophilicity of ethane seems to be the reason for the barrier height lowering through ~ 3.3 to 3.7 kcal/mol for $^1\text{CHCl}$ insertion. Similarly, the barrier height for $^1\text{CCl}_2$ into ethane does show the same trend (a reduction through ~ 4.5 to 5.9 kcal/mol). This aspect is clear from the fact that the reactivity decreases as the chlorination increases in the carbene. An approximately two to five fold increase in activation barrier (at B3LYP and MP2 levels) for inverted σ/π approach of $^1\text{CHCl}/^1\text{CCl}_2$ may be attributed to the fact that the migrating hydrogen approaches the smaller lobe of the σ orbital of carbene carbon having the lone pair of electrons. Moreover the predominant π -donation over σ -attraction by chlorine atom could be the reason for this enormous increase in the activation barrier with the degree of chlorination.

Over all, the σ_{CH_2} orientation for $^1\text{CHCl}(^1\text{CCl}_2)$ insertion has been found to be slightly preferred over the π_{CH_2} approach. The relevant geometrical parameters of the transition states for the $^1\text{CHCl}$ and $^1\text{CCl}_2$ insertions in σ and π orientations have been shown in Fig. 1. The TS for $^1\text{CCl}_2$ insertion (TS-2) comes much later along the reaction coordinate than that for $^1\text{CHCl}$ insertion (TS-1) as reflected in the relative C2-H1 bond distances of 1.430 (1.345) and 1.274 (1.202) Å and the charges on H1 of 0.284(0.260) and 0.253 (0.225) respectively.

3.2 Singlet Chlorocarbenes Insertion into C-H of Higher Alkanes

The higher alkanes have three types of C-H bonds (primary, secondary and tertiary; Fig. 1) into which the carbenes can be inserted *via* the σ/π approach. The B3LYP and MP2 barrier heights for these have been listed in Table 1. For the σ -insertion of $^1\text{CHCl}$ into primary C-H of ethane to *iso*-butane, the barrier heights calculated at B3LYP/6-31g (d, p) have been found to be around 2.0 kcal/mol. But MP2 predictions (~ 3.9 kcal/mol) are slightly higher by *ca.* 1.9 kcal/mol. The small reduction in barrier height on moving from methane to ethane (3.5 kcal/mol) should have been due to the enhanced nucleophilicity by the methyl substituent in methane and this effect is insignificant on further increasing the hydrocarbon chain length. The barrier heights were found to be negative for σ approach at B3LYP/6-31g(d, p) for the insertion at secondary and tertiary C-H (Table 1). But at MP2/6-31g(d, p) positive values have been observed, exception being the C2 of *iso*-butane. The enhanced nucleophilicity of the substrate might be the reason for the negative barrier heights.

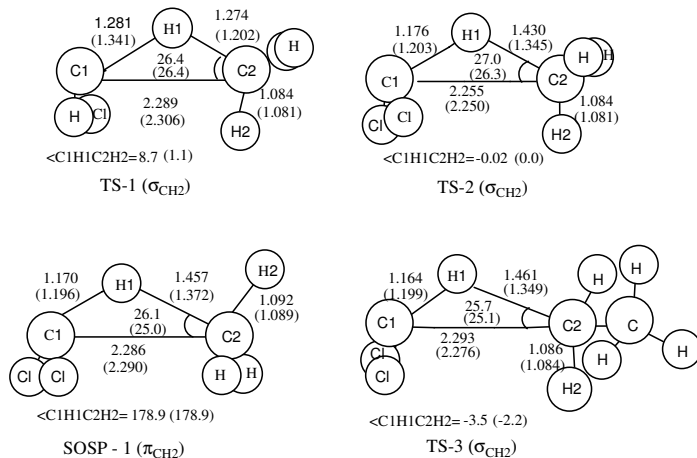


Fig. 1. Selected optimized geometrical parameters (distances in Å and angles in degrees) of the transition states at B3LYP/6-31g (d, p) and MP2/6-31g (d, p) (in parentheses) levels

If the carbene is $^1\text{CCl}_2$ the barrier height decreases to ~ 16.8 kcal/mol at DFT for all primary C-H insertions on comparison with that for methane and it falls around 13.7 kcal/mol at MP2 level. The decreased electrophilicity may be the reason for the enhanced barrier heights. This trend also reveals that the steric factor has no noticeable influence upon the insertion pathway. For secondary C-H, the barrier heights have been found to be reduced to an average value of *ca.* 0.78 kcal/mol and 8.27 kcal/mol for insertion of $^1\text{CHCl}$ and $^1\text{CCl}_2$ respectively at MP2 level. The proximity of the electron donating alkyl group to the insertion site might be the reason for this result. This is further supported by the lowered barrier heights for insertion of $^1\text{CHCl}$ or $^1\text{CCl}_2$ into the tertiary C-H of *iso*-butane both at B3LYP and MP2 levels.

3.3 Energetics

In the case of insertion of $^1\text{CHCl}$ and $^1\text{CCl}_2$ into an alkane, the order of the activation barrier is : tertiary < secondary < primary (Table 1). The above trend draws support from the fact that the pair of electrons on the carbene carbon involved in the bonding process is more and more stabilized with the degree of chlorination. Due to the less availability of the electron pair on the carbene carbon, ease of bond formation is inhibited. The NBO [33] analyses quantify this aspect in terms of the energies of the electron pairs on $^1\text{CHCl}$ and $^1\text{CCl}_2$ as -0.4057 and -0.5595 au at B3LYP and -0.4535 and -0.6019 au at MP2 respectively.

The enthalpies of chlorocarbene insertion reactions calculated from the equation,

$$\Delta_r H^0(298 \text{ K}) = \sum (\epsilon_o + H_{\text{corr}})_{\text{products}} - \sum (\epsilon_o + H_{\text{corr}})_{\text{reactants}}$$

ϵ_o - total electronic energy; H_{corr} - correction to the enthalpy due to internal energy have been listed in Table 2. All the reaction enthalpies show the exothermicity of the insertion reactions indicating that all the transition states analyzed resemble the reactants rather than the products [34]. The proximity of the transition states to the

reactants deviates with the degree of chlorination of carbene. HF/6-31g(d, p) calculations indicate an early transition state for the $^1\text{CH}_2$ with exothermicity around -94 kcal/mol [17] compared to those formed by $^1\text{CHCl}/^1\text{CCl}_2$. Irrespective of the level of theory (B3LYP or MP2) followed, the insertions of $^1\text{CHCl}$ form the transition states earlier than those of $^1\text{CCl}_2$ insertions as revealed by exothermicity values (Table 2).

Table 2. Heat of reaction (kcal/mol) for insertion of singlet chlorocarbenes into C-H bonds of alkanes using 6-31G(d,p) at B3LYP and MP2 levels

alkane	CHCl		CCl ₂	
	B3LYP	MP2	B3LYP	MP2
methane	-77.72	-94.82	-62.92	-78.25
ethane	-79.31	-97.63	-64.48	-81.38
propane (C1)	-79.14	-97.66	-64.29	-81.48
propane (C2)	-79.83	-99.75	-63.89	-82.79
n-butane(C1)	-79.14	-97.73	-64.31	-81.59
n-butane(C2)	-78.73	-99.09	-62.47	-82.88
iso-butane(C1)	-78.07	-97.0	-63.41	-78.54
iso-butane(C2)	-79.25	-101.1	-62.99	-84.18

3.4 Transition State Geometries (NBO and IRC Analyses)

The relevant computed data for all the transition states have been shown in Table 3 and Table 4. A scrutiny of the bond breaking and bond formation steps corresponding to C2-H1 and C1-H1 respectively during the insertion process reveals that it is a concerted reaction. It is observed that the maturity of C1-H1 bond takes place earlier than the C1-C2 bond in the TSs (Table 3 and Table 4). The C1-H1 distances are inversely related to the barrier heights as expected. A similar relationship is also obvious in the C1-C2 distances (Table 3 and Table 4). The C2-H1 distances in the bond breaking process in all the transition states for $^1\text{CHCl}$ insertions at B3LYP (MP2)/6-31g(d, p) have been found to be *ca.* 1.264(1.197) Å, 1.272(1.149) Å and 1.324(1.132) Å, respectively for primary, secondary and tertiary C-H of higher alkanes. This shows again the belated transition states compared to that of $^1\text{CH}_2$ insertions [17]. The mean glide angle (θ_{H1C2C1}) decreases from primary to tertiary C-H of the alkane (Table 3) and it gets reflected in the corresponding activation barriers (Table 1). Further correlation is also noticed when the torsion angles ϕ_{C1H1C2H2} have been analyzed (Table 3). The quantum of charge transfer from alkane to carbene in TSs increases with the increase in the number of alkyl substituents on the C2 carbon (Table 3). It shows the inverse relationship between the barrier height and quantum of charge transfer. The same trend is observed in the case of $^1\text{CCl}_2$.

The NBO analyses of charge distribution in the transition state species give some insight into the insertion reactivity. They show that the $\sigma_{\text{CH}} \rightarrow \text{LP}^*$ interaction between the C-H bond of alkane and LP^* of the carbene carbon seems to give the

Table 3. Selected geometrical parameters (distances in Å, angles in degrees, and barriers in kcal/mol) at the TSs of $^1\text{CHCl}$ with alkanes at B3LYP (MP2)/6-31g(d, p)

alkane	r_{C1H1}	r_{C1C2}	r_{C2H1}	θ_{H1C1C2}	θ_{H1C2C1}	Φ_{C1H1C2H2}	q_{ct}
methane	1.281	2.289	1.274	26.2	26.4	8.7	0.237
	(1.341)	(2.306)	(1.202)	(23.5)	(26.4)	(1.1)	(0.193)
ethane	1.296	2.343	1.260	23.2	23.9	12.0	0.2527
	(1.409)	(2.401)	(1.173)	(19.6)	(23.8)	(9.8)	(0.161)
propane(c1)	1.295	2.340	1.261	23.4	24.1	9.5	0.2531
	(1.416)	(2.406)	(1.172)	(19.6)	(23.9)	(4.6)	(0.157)
propane(c2)	1.290	2.421	1.272	18.9	19.2	43.2	0.268
	(1.536)	(2.545)	(1.143)	(15.6)	(21.2)	(5.3)	(0.105)
n-butane(c1)	1.296	2.342	1.261	23.3	24.0	8.7	0.253
	(1.422)	(2.412)	(1.170)	(19.4)	(23.8)	(3.9)	(0.154)
n-butane(c2)	1.290	2.391	1.271	20.8	21.1	38.7	0.271
	(1.491)	(2.521)	(1.154)	(15.5)	(20.2)	(31.9)	(0.124)
isobutane(c1)	1.289	2.334	1.266	23.8	24.3	10.0	0.259
	(1.425)	(2.421)	(1.168)	(18.9)	(23.4)	(7.8)	(0.152)
isobutane(c2)	1.235	2.358	1.324	23.7	22.1	8.8	0.309
	(1.614)	(2.690)	(1.132)	(9.7)	(13.8)	(18.4)	(0.079)

q_{ct} – quantum of charge transfer from alkane to carbene at the TSs.

Table 4. Selected geometrical parameters (distances in Å, angles in degrees, and barriers in kcal/mol) at the TSs of $^1\text{CCl}_2$ with alkanes at B3LYP (MP2)/6-31g(d,p)

alkane	r_{C1H1}	r_{C1C2}	r_{C2H1}	θ_{H1C1C2}	θ_{H1C2C1}	Φ_{C1H1C2H2}	q_{ct}
methane	1.176	2.255	1.430	33.5	27.0	0.02	0.297
	(1.203)	(2.250)	(1.345)	(29.7)	(26.3)	(0.0)	(0.316)
ethane	1.164	2.293	1.461	33.0	25.7	-3.5	0.337
	(1.199)	(2.276)	(1.349)	(28.5)	(25.1)	(2.2)	(0.353)
n-propane(c1)	1.164	2.291	1.461	33.1	25.8	0.7	0.336
	(1.201)	(2.278)	(1.348)	(28.4)	(25.1)	(0.1)	(0.352)
n-propane(c2)	1.143	2.413	1.532	29.9	21.8	1.1	0.354
	(1.187)	(2.379)	(1.380)	(23.9)	(20.4)	(3.3)	(0.375)
n-butane(c1)	1.163	2.295	1.470	33.3	25.8	4.0	0.342
	(1.199)	(2.277)	(1.350)	(28.5)	(25.1)	(2.9)	(0.361)
n-butane(c2)	1.141	2.415	1.546	30.6	22.1	2.5	0.358
	(1.184)	(2.375)	(1.389)	(24.6)	(20.8)	(3.9)	(0.385)
isobutane(c1)	1.163	2.291	1.462	33.1	25.7	0.2	0.348
	(1.199)	(2.277)	(1.350)	(28.5)	(25.1)	(1.2)	(0.357)
isobutane(c2)	1.130	2.476	1.594	29.5	20.5	0.07	0.374
	(1.179)	(2.401)	(1.392)	(22.8)	(19.2)	(0.01)	(0.404)

q_{ct} – quantum of charge transferred from alkane to carbene at the TSs.

strongest stabilization. Finally we observed that there was a net charge flow from the alkane unit to the inserting carbene moiety. The quantum of charge transfer from alkane to carbene supporting the donor-acceptor interaction in the transition states for all the insertion reactions have been collected in Tables 3 and 4. The inverse relationship between the quantum of charge transfer and the activation barriers reveals the fact that for the favorable insertion, the nucleophilicity of the alkane should have been enhanced either sterically or electronically. This correlation holds good for all the reactions analysed in this investigation. For example, in the case of ${}^1\text{CCl}_2$ insertion

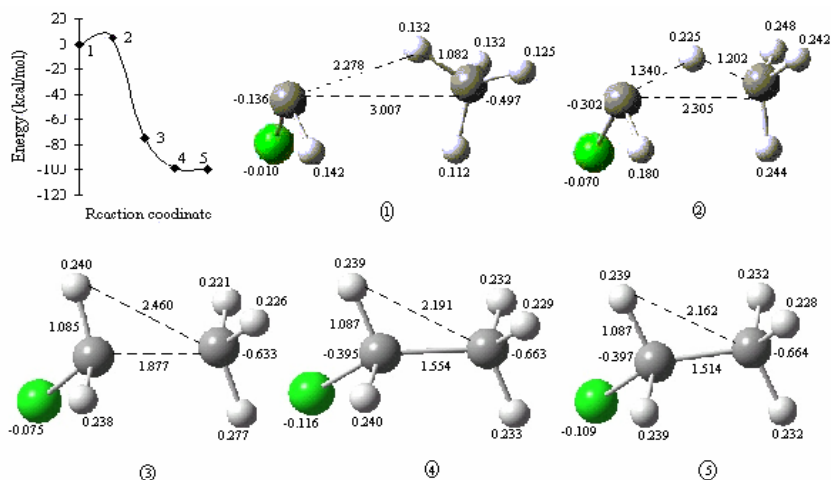


Fig. 2. IRC of insertion of ${}^1\text{CHCl}$ into C-H bond of methane, calculated at MP2/6-31G(d, p)

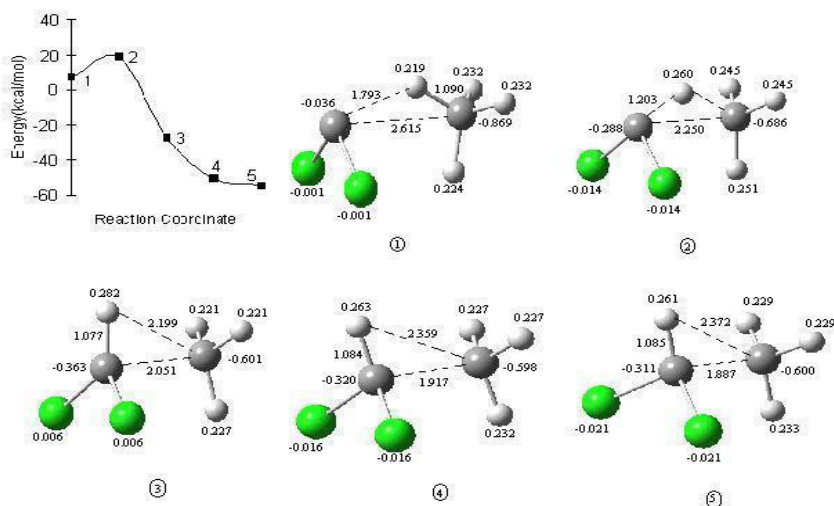


Fig. 3. IRC of insertion of ${}^1\text{CCl}_2$ into C-H bond of methane, calculated at MP2/6-31G(d, p)

into *iso*-butane at MP2 level the barrier heights, 11.83 and 4.42 kcal/mol, for the primary and tertiary C-H respectively correlate with the charge transfer of 0.357 and 0.404.

The reaction path for the insertion of $^1\text{CHCl}$ into C-H of methane has been studied by the IRC method at MP2 level (Fig. 2). In the transition state C2-H1 bond is elongated by 10.9%. The stretching of the C2-H1 bond resulting in the C1-H1 bond formation and the charge transfer from the methane to chlorocarbene are closely linked together. Moreover the C1-H1 and the C1-C2 bond formations are found to be asynchronous.

The same trend is followed also in the case of $^1\text{CCl}_2$ (Fig. 3), the difference being that the less electrophilic character of $^1\text{CCl}_2$ compared to $^1\text{CHCl}$ results in a significantly reduced reactivity which is reflected in the higher activation barrier.

4 Summary

The singlet carbene insertion into the alkanes at the primary, secondary and tertiary C-H bonds has been analyzed and the influence of chlorine on the transition states, energetics, geometrical parameters etc., has been investigated both at B3LYP and MP2 levels. Both the theories predict that the activation barrier is a function of the degree of chlorination of carbene and the type of C-H into which insertion occurs. Also, among the two different types of carbene moiety approaches, i.e., σ and π , the σ approach is preferred over the π approach at both levels of theory as the π approach leads to the eclipsed conformation which is a second order saddle point. Intrinsic reaction coordinate analysis for the insertion in the case of $^1\text{CHCl}$ and $^1\text{CCl}_2$ into the primary C-H of methane at MP2 has been carried out to show that the C1-H1 bond is formed first followed by the C1-C2 bond. The NBO analyses have been done with a view to analyzing the charge transfer processes during the insertion reactions.

References

1. Drake, S.A., Standard, J.M., Quandt, R.W.: J. Phys. Chem.A. 106(2002) 1357
2. Kirmse, W.: Carbene Chemistry. 2nd Edn. Academic Press, New York (1971)
3. Fischer, E.O., Maasbol, A.: Angew. Chem., Int. Ed. Engl. 3 (1964) 580
4. Brahm, D.L.S., Dailey, W.P.: Chem. Rev. 96 (1996) 1585
5. Storer, J.W., Houk, K.N.: J. Am. Chem. Soc. 115 (1993) 10426
6. Gleiter, R., and Hoffmann, R.: J. Am. Chem. Soc. 90 (1968) 1475
7. Irikura, K., Goddard III, W.A., Beauchamp, J.L.: J. Am. Chem. Soc. 114 (1992) 48
8. Russon, N., Sicilia, E., Toscano, M.: J. Chem. Phys. 97 (1992) 5031
9. von W., Doering, E., Prinzbach, H.: Tetrahedron 6 (1959) 24
10. Bauchlicher, C., Bicher, W., Haber, K., Schaefer, H.F., Bender III, C.F.: J. Am. Chem. Soc. 99 (1977) 3610
11. Krishnan, R., Pople, J.A.: Int. J. Quantum Chem. Symp. 14 (1980) 91
12. Gordon, M.S., Gano, D.R.: J. Am. Chem. Soc. 106 (1984) 5421
13. Jones, M., Moss, R.A.: Carbenes, Vol.1, Wiley, New York (1972)
14. Gordon, M.S., Boatz, J.A., Gano, D.R.: J. Am. Chem. Soc. 109 (1987) 1323
15. Gano, D.R., Gordon, M.S., Boatz, J.A.: J. Am. Chem. Soc. 113 (1991) 6711

16. Gordon, M.S., Gano, D.R., Binkley, J.S., Frisch, M.J.: *J. Am. Chem. Soc.* 108 (1986) 2191
17. Ramalingam, M., Ramasami, K., Venuvanalingam, P., Sethuraman, V.: *J. Mol. Struct.(Theochem)* 755 (2005) 169
18. Zuev, P., Sheridan, R.S.: *J. Am. Chem. Soc.* 116 (1994) 4123
19. Arduengo, J., Goerlich, J.R., Krafczyk, R., Marshall, W.J.: *Angew. Chem., Int. Ed., Engl.* 37 (1998)1963
20. Jorgensen, W.L., Salem, L.: *The Organic Chemist's Book of Orbitals*. Academic Press, New York (1973)
21. Meredith, C., Hamilton, T.P., Schaefer III, H.F.: *J. Phys. Chem.* 96 (1992) 9250
22. Gaussian 03, Revision C.02, Gaussian, Inc., Wallingford CT (2004)
23. Lee, C., Yang, W., Parr, R.G.: *Physical Review B* 37 (1988) 785
24. Becke, D.: *Phys. Rev. A* 38 (1988) 3098
25. Miehlich, B., Savin, A., Stoll, H., Preuss, H.: *Chem. Phys. Lett.* 157 (1989) 200
26. Becke, A.D.: *J. Chem. Phys.* 98 (1993) 5648
27. Becke, A.D.: *J. Chem. Phys.* 104 (1996) 1040
28. Franel, M.M., Pietro, W.J., Hehre, W.J., Bimcley, J.S., Gordon, M.S., DeFrees, D.J., Pople, J.A.: *J. Chem. Phys.*, 77 (1982) 3654
29. Hariharan, P.C., Pople, J.A.: *Chem. Phys. Lett.* 66 (1972) 217
30. Bach, R.D., Su, M.D., Aldabbagh, E., Andres, J.L., Schlegel, H.B.: *J. Am. Chem. Soc.* 115 (1993) 10237
31. Bach, R.D., Andres, J.L., Su M.D., McDouall, J.J.W.: *J. Am. Chem. Soc.* 115 (1993) 5768
32. Gilles, M.K., Lineberger, W.C., Ervin, K.M.: *J. Am. Chem. Soc.* 115 (1993) 1031
33. Glendening, E.D., Reed, A.E., Carpenter, J.E., Weinhold, F., Curtiss, L.A.: *Chem. Rev.* 88 (1988) 899. NB Version 3.1
34. Carpenter, J.E.: Ph.D. Thesis, University of Wisconsin (Madison, WI) (1987)

Theoretical Gas Phase Study of the Gauche and Trans Conformers of 1-Fluoro-2-Haloethanes $\text{CH}_2\text{F}-\text{CH}_2\text{X}$ ($\text{X}=\text{Cl}, \text{Br}, \text{I}$) by Ab Initio and Density Functional Methods: Absence of Gauche Effect

Ponnadurai Ramasami

Faculty of Science, Department of Chemistry,
University of Mauritius, Réduit, Mauritius
p.ramasami@uom.ac.mu
www.pages.intnet.mu/ramasami/

Abstract. This is a systematic theoretical gas phase study of the gauche and trans conformers of 1-fluoro-2-haloethanes ($\text{FCH}_2\text{CH}_2\text{X}$, $\text{X}=\text{Cl}, \text{Br}$ and I). The methods used are second order Møller-Plesset theory (MP2) and density functional theory (DFT). The basis set used is 6-311++(d,p) for all atoms except that 3-21G and CRENBL ECP are used for iodine atom. The functional used for DFT method is B3LYP. G2/MP2 calculation has also been carried out using MP2 optimised structure. The results indicate that unlike 1,2-difluoroethane, there is the absence of gauche effect and thus there is more preference for the trans conformer. The preference for the more stable trans conformer increases with increasing atomic size of the substituted halogen atom. The same trend is observed for energy difference between the gauche and trans conformers. The 1-fluoro-2-haloethanes have also been subjected to vibrational analysis.

1 Introduction

1,2-Disubstituted ethanes are amongst the simplest molecules showing conformational isomerism due to restricted rotation about carbon-carbon single bond [1]. Elemental analysis suggests that the 1,2-disubstituted ethanes can exist as a gauche and a trans conformer. It is generally found that the trans conformer is more stable than the gauche form and this is due to steric hindrance in the gauche conformation [2]. However fluorine has strong stereoelectronic and polar effects on the conformers of fluorine substituted molecules [3-8]. The net effect is that the gauche conformer is unusually more stable for the fluoro substituted conformers and this has been associated with the “gauche effect” [2,9]. To put this into context of this work, gauche-1,2-difluoroethane predominates in the gas phase and this has been found both experimentally [10-12] and from theoretical calculations [1,13-15]. It is generally agreed that the gauche conformer of 1,2-difluoroethane is more stable than the trans conformer by about 4.18 kJ/mol.

Although 1-fluoro-2-chloroethane has been the target [15] for conformational studies, 1-fluoro-2-bromoethane and 1-fluoro-2-iodoethane have not been explored. The scope of theoretical conformational studies has increased in recent years with the

availability of fast computers and quantum mechanical software. In order to provide more understanding to the gauche and trans conformers of 1,2-disubstituted-haloethanes and in continuation with the previous work [16], this theoretical study aims to explore the gauche and trans conformers, figure 1, of 1-fluoro-2-haloethane ($\text{FCH}_2\text{CH}_2\text{X}$, $\text{X}=\text{Cl}$, Br and I) in the gas phase. These conformers have been studied with a view to obtain (i) the optimised structural parameters, (ii) the energy difference between the gauche and trans conformers or the torsional barrier and (iii) related thermodynamics properties for torsional rotation. Apart from energy calculations, the conformers of 1-fluoro-2-haloethanes have also been subjected to vibrational analysis. The results of this study are reported herein.

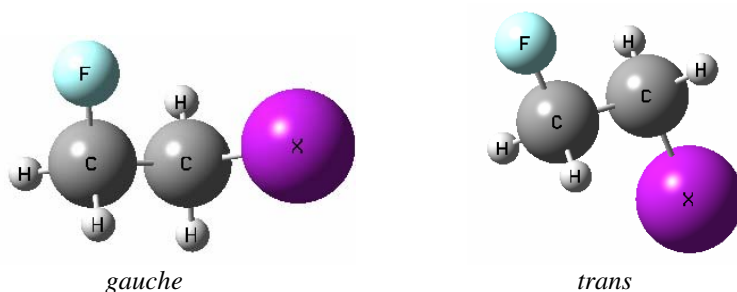


Fig. 1. Gauche and trans conformers of 1-fluoro-2-haloethane

2 Calculations

In this study all the calculations have been carried out using Gaussian 03W [17] program. GaussView 3.0 [18] has been used for visualising the molecules.

The calculations have been carried out using second order Møller-Plesset perturbation theory (MP2) and density functional theory (DFT). The basis set used is 6-311++G(d,p) for all atoms except that 3-21G and CRENBL ECP [19] are used for iodine only. The basis set 6-311++G(d,p) is appropriate for conformational studies for small molecules [15,16]. The functional used for DFT method is B3LYP. A conformer has first been optimised and the optimised structure has then been used for frequency calculations using the same method and basis set involved for optimisation. G2/MP2 calculation has also been carried out but could not be done for 1-fluoro-2-iodoethane due to unavailability of basis sets used by the method for iodine atom. Optimised structure at MP2/6-311++G(d,p) level has been used for G2/MP2 calculation.

3 Results and Discussion

The optimised structural parameters for the trans and gauche conformers of the 1-fluoro-2-haloethanes which are of interest and obtained using the MP2 and B3LYP methods and the basis set described in the calculations section are summarised in Table 1. The available literature values for the gauche conformer of 1-fluoro-2-chloroethane are also

included and it can be found that there is a good comparison. It is interesting to note that there is little difference between the MP2 and B3LYP calculated parameters. However bond lengths prediction from B3LYP are generally larger than MP2 values. The angles predicted by the two methods show more of random variations for a general statement.

Table 1. Optimised structural parameters for the 1-fluoro-2-haloethanes

Parameter	MP2		B3LYP	
	Trans	Gauche	Trans	Gauche
1-fluoro-2-chloroethane				
r (C-F)	1.392 Å	1.388 Å (1.365 Å)*	1.402 Å	1.395 Å
r (C-C)	1.515 Å	1.509 Å (1.530 Å)*	1.517 Å	1.509 Å
r (C-Cl)	1.779 Å	1.778 Å (1.787 Å)*	1.811 Å	1.810 Å
∠ (CCF)	108.3°	111.5° (109.7°)*	107.9°	110.8°
∠ (CCCl)	108.7°	110.2° (111.1°)*	109.2°	112.6°
τ (FCCCl)	180.0°	67.2° (68.3°)*	180.0°	69.8°
1-fluoro-2-bromoethane				
r (C-F)	1.394 Å	1.387 Å	1.404 Å	1.394 Å
r (C-C)	1.515 Å	1.510 Å	1.515 Å	1.508 Å
r (C-Br)	1.940 Å	1.940 Å	1.972 Å	1.974 Å
∠ (CCF)	108.1°	110.4°	107.9°	111.0°
∠ (CCBr)	109.1°	112.0°	109.2°	112.8°
τ (FCCBr)	180.0°	66.5°	180.0°	68.6°
1-fluoro-2-iodoethane				
r (C-F)	1.392 Å	1.386 Å	1.402 Å	1.393 Å
r (C-C)	1.506 Å	1.504 Å	1.513 Å	1.508 Å
r (C-I)	2.148 Å	2.150 Å	2.190 Å	2.193 Å
∠ (CCF)	108.4°	110.6°	108.0°	111.0°
∠ (CCI)	110.5°	113.5°	110.0°	113.6°
τ (FCCI)	180.0°	64.5°	180.0°	67.1°

* Reference 9

The energies of the trans and gauche conformers of the 1-fluoro-2-haloethanes calculated at MP2 and B3LYP and basis sets described in the calculations section are given in Table 2. These energies have been obtained after full geometry optimisation which has been verified by frequency calculation. G2/MP2 energies are also given in Table 2. The torsional barrier and related thermodynamic parameters are summarized in Table 2. A glance at Table 2 clearly shows that the trans conformer is more stable than the gauche form unlike the behaviour of 1,2-difluoroethane. The energies of all the conformers are more negative using the DFT calculation and thus this means that conformers should be more stable than on the basis of DFT rather than MP2 calculations. The rotational barrier becomes larger with increasing atomic size of the substituted halogen atom. The rotational barriers predicted using MP2 calculations are greater than DFT calculations but they are all within acceptable deviations. The results for 1-fluoro-2-haloethanes are in agreement with literature values [9,20]. The free energy barrier has been used to estimate the relative percentages of the trans and gauche conformers. It is found that at 298 K, the percentages of the trans conformer calculated using MP2 values are 70, 80 and 86 respectively for X=Cl, Br and I (FCH₂CH₂X). These percentages are of the same order based on DFT calculations.

Table 2. Calculated energies and thermodynamic parameters of the 1-fluoro-2-haloethanes

Compound	Trans (Hartrees)	Gauche (Hartrees)	Torsional barrier (kJ/mol)	ΔG (298 K) (kJ/mol)
MP2/6-311++G(d,p) for all atoms except 3-21G for iodine atom				
1-fluoro-2-chloroethane	637.7023655	637.7014765	2.33	2.10
1-fluoro 2-bromoethane	2750.5506970	2750.5492520	3.79	1.81
1-fluoro 2-iodoethane	7065.3805704	7065.3785745	5.24	4.41
MP2/6-311++G(d,p) for all atoms except CRENBL ECP for iodine atom				
1-fluoro 2-iodoethane	287.7294325	287.7274743	5.14	4.65
B3LYP/6-311++G(d,p) for all atoms except 3-21G for iodine atom				
1-fluoro-2-chloroethane	638.7430021	638.7422184	2.06	3.46
1-fluoro 2-bromoethane	2752.6650578	2752.6637633	3.40	2.96
1-fluoro 2-iodoethane	7068.5253779	7068.5235271	4.86	4.01
B3LYP/6311++G(d,p) for all atoms except CRENBL ECP for iodine atom				
1-fluoro 2-iodoethane	289.9790627	289.9771779	4.95	4.40
G2/MP2				
1-fluoro-2-chloroethane	637.9201750	637.9193236	2.24	2.24
1-fluoro 2-bromoethane	2750.7580730	2750.7566020	3.86	3.83

The trans and gauche conformers of the 1-fluoro-2-haloethanes have also been subjected to vibrational analysis. All the 18 fundamentals of the trans and gauche

conformers have been assigned. The calculated frequencies are reported in Tables 3 and 4. These uncorrected frequencies have been assigned after visualization using GaussView. The values indicate that predictions with MP2 level of theory are systematically larger than B3LYP level of theory. The calculated frequencies for 1-fluoro-2-chloroethane are in agreement with literature values obtained theoretically [9] using B3LYP/6-311+G(d,p) and experimentally [21].

Table 3. Calculated frequencies (cm^{-1}) of the 1-fluoro-2-haloethanes and their assignments obtained at MP2/6-311+G(d,p) for all atoms except 3-21G for iodine atom

1-fluoro -2-chloroethane		1-fluoro -2-bromoethane		1-fluoro -2-iodoethane		Assignment
Trans	Gauche	Trans	Gauche	Trans	Gauche	
138.7 (10.3)	140.3 (2.3)	133.9 (8.9)	122.6 (1.9)	135.6 (6.9)	108.8 (2.2)	FCCX torsion
247.8 (12.3)	294.8 (0.9)	220.7 (7.5)	269.4 (1.6)	190.2 (4.4)	243.2 (1.2)	XCC bend
395.4 (3.0)	476.1 (13.0)	343.0 (5.3)	457.4 (12.9)	320.1 (7.1)	441.9 (10.0)	FCC bend
817.1 (0.2)	731.5 (24.7)	722.8 (25.8)	615.6 (10.6)	681.2 (14.6)	555.5 (0.9)	C-X str
827.5 (38.4)	872.9 (13.0)	800.4 (0.8)	849.7 (6.3)	786.5 (0.7)	818.8 (3.5)	CH ₂ rock
1066.8 (106.9)	1005.9 (7.6)	1043.0 (3.5)	982.2 (8.7)	1028.3 (2.8)	995.4 (24.1)	CH ₂ wag
1101.4 (2.4)	1087.9 (45.1)	1056.4 (114.2)	1054.7 (33.6)	1047.4 (122.3)	1046.7 (26.0)	C-F str
1113.0 (23.7)	1132.3 (44.3)	1106.4 (33.5)	1126.9 (58.2)	1126.4 (38.2)	1133.4 (53.3)	C-C str
1240.1 (0.9)	1248.9 (1.4)	1236.1 (1.0)	1226.7 (2.5)	1255.1 (0.9)	1245.9 (2.1)	CH ₂ twist
1326.2 (0)	1301.2 (0.6)	1282.3 (16.4)	1283.1 (2.9)	1309.8 (21.7)	1283.3 (1.8)	CH ₂ twist
1333.4 (14.5)	1381.2 (30.2)	1322.8 (0)	1344.0 (41.1)	1338.0 (0)	1364.5 (51.3)	CH ₂ wag
1447.8 (3.6)	1452.3 (12.2)	1436.7 (5.4)	1446.2 (12.6)	1433.8 (5.5)	1430.1 (1.8)	CH ₂ wag
1510.0 (5.4)	1477.6 (10.6)	1506.3 (6.2)	1474.1 (11.7)	1488.7 (12.7)	1470.3 (25.2)	CH ₂ scissor
1532.8 (0.8)	1517.1 (2.4)	1532.5 (1.0)	1519.0 (2.5)	1537.7 (1.8)	1519.1 (1.3)	CH ₂ scissor
3119.3 (14.8)	3099.0 (24.2)	3118.6 (13.7)	3093.7 (25.9)	3119.8 (18.9)	3098.8 (34.0)	CH ₂ sym str
3140.8 (17.8)	3124.6 (16.9)	3143.2 (11.0)	3128.5 (10.6)	3175.4 (9.4)	3156.4 (5.8)	CH ₂ sym str
3184.4 (9.8)	3170.5 (16.3)	3185.0 (11.1)	3166.8 (15.5)	3187.0 (17.7)	3170.6 (24.4)	CH ₂ antisym str
3215.8 (9.4)	3196.7 (6.4)	3220.3 (4.4)	3202.5 (2.6)	3257.3 (2.3)	3237.9 (2.2)	CH ₂ antisym str

Values in bracket are infrared intensities in (km/mol), (X=Cl, or Br or I).

Table 4. Calculated frequencies (cm^{-1}) of the 1-fluoro-2-haloethanes and their assignments obtained at B3LYP/6-311++G(d,p) for all atoms except 3-21G for iodine atom

1-fluoro -2-chloroethane		1-fluoro -2-bromoethane		1-fluoro -2-iodoethane		Assignment
Trans	Gauche	Trans	Gauche	Trans	Gauche	
125.1 (11.1)	129.6 (2.5)	123.8 (9.4)	111.9 (1.9)	127.3 (8.0)	99.1 (2.0)	FCCX torsion
241.9 (13.3)	285.4 (1.2)	212.1 (8.0)	258.8 (2.1)	188.9 (5.6)	241.5 (1.8)	XCC bend
376.9 (2.7)	461.5 (14.7)	327.3 (5.1)	444.3 (15.6)	295.7 (5.2)	427.7 (14.6)	FCC bend
753.6 (50.9)	662.0 (34.0)	672.8 (36.8)	565.9 (15.8)	620.1 (29.2)	496.7 (4.3)	C-X str
797.9 (0.8)	848.0 (15.3)	785.1 (1.6)	828.1 (7.8)	758.4 (1.1)	794.3 (4.0)	CH ₂ rock
1017.9 (131.1)	968.9 (9.4)	1006.2 (142.1)	952.4 (12.3)	979.8 (1.5)	951.4 (21.7)	CH ₂ wag
1061.5 (1.9)	1046.6 (58.5)	1010.8 (2.8)	1019.5 (42.2)	1001.2 (149.3)	1002.4 (28.1)	C-F str
1069.8 (11.4)	1089.6 (40.7)	1068.1 (18.8)	1085.1 (57.8)	1076.9 (27.1)	1090.3 (62.5)	C-C str
1211.4 (0.9)	1215.2 (2.5)	1206.4 (1.0)	1199.2 (3.3)	1219.7 (0.9)	1205.0 (3.8)	CH ₂ twist
1290.1 (12.3)	1270.8 (0.28)	1246.4 (16.7)	1250.1 (6.4)	1230.9 (25.2)	1235.7 (16.0)	CH ₂ twist
1297.6 (0)	1335.6 (33.6)	1295.9 (0)	1308.0 (44.1)	1305.0 (0)	1302.8 (54.4)	CH ₂ wag
1409.5 (2.6)	1416.4 (10.9)	1402.5 (3.8)	1413.8 (12.3)	1405.3 (7.6)	1420.4 (11.4)	CH ₂ wag
1492.4 (4.0)	1462.2 (9.7)	1488.4 (5.0)	1458.3 (10.6)	1477.7 (6.3)	1444.0 (13.4)	CH ₂ scissor
1510.5 (2.5)	1489.3 (4.1)	1509.3 (2.4)	1490.2 (4.2)	1510.3 (1.7)	1494.9 (3.1)	CH ₂ scissor
3063.5 (16.4)	3034.7 (25.3)	3065.4 (14.9)	3028.5 (26.3)	3065.5 (17.5)	3025.2 (30.6)	CH ₂ sym str
3095.8 (16.2)	3081.5 (11.6)	3105.1 (9.8)	3090.0 (6.0)	3118.7 (17.8)	3091.4 (14.2)	CH ₂ sym str
3116.2 (12.4)	3098.7 (20.2)	3119.1 (13.3)	3099.3 (19.0)	3127.0 (9.1)	3112.9 (13.7)	CH ₂ antisym str
3160.8 (8.1)	3143.7 (5.4)	3173.9 (3.3)	3158.0 (2.1)	3200.4 (2.7)	3181.4 (2.3)	CH ₂ antisym str

Values in bracket are infrared intensities in (km/mol), (X=Cl, or Br or I).

4 Conclusions

The objectives of this work using theoretical methods, have been achieved with the determination of the optimised structural parameters, torsional barrier and related thermodynamics parameters for the 1-fluoro-2-haloethanes. The results indicate a preference for the trans conformer and hence the absence of gauche effect. The latter is common for fluorine substituted conformers such as 1,2-difluoroethane. It can also be deduced that there is increasing preference for the trans conformer with increasing

size of the substituted halogen atom. The results of this study are also in agreement with literature findings for 1,2-disubstituted ethanes and thus they are promising for the modelling of conformers of these molecules by theoretical method.

Acknowledgements

The author acknowledges facilities from the University of Mauritius and anonymous reviewers for their comments to improve the manuscript.

References

1. Dixon D.A., Matsuzawa N., Walker S.C.: Conformational Analysis of 1,2-Dihaloethanes: A Comparison of Theoretical Methods. *J. Phys. Chem.* 96 (1992) 10740-10746
2. Wolfe S.: The Gauche Effect. Some Stereochemical Consequences of Adjacent Electron Pairs and Polar Bonds. *Acc. Chem. Res.* 5 (1972) 102-111
3. O'Hagan D., Rzepa H.S.: Some Influences of Fluorine in Bioorganic Chemistry. *Chem. Commun.* (1997) 643-652
4. Rablen P.R., Hoffman R.W., Hrovat D.A.: Is Hyperconjugation Responsible for the "Gauche Effect" in 1-fluoropropane and 1-fluoroethanes?. *J. Chem. Soc. Perkin Trans. 2.* 8 (1999) 1719-1726
5. O'Hagan D., Bilton C., Howard J.A.K., Knight L., Tozer D. J.: The Preferred Conformation of N- β -fluoroethylamides. Observation Fluorine Amide Gauche Effect. *J. Chem. Soc. Perkin. Trans. 2.* 8 (2000) 600-607
6. Tavasli M., O'Hagan D., Pearson C. Petty M.C.: The Fluorine Gauche Effect. *Langmuir Isotherms Reprot the Relative Conformational Stability of (+/-)-Erythro- and (+/-)-Threo-9,10-Difluorostearic acids.* *Chem. Commun.* 7 (2002) 1226-1227
7. Briggs C.R., Allen M.J., O'Hagan D., Tozer D.J., Slawin A.M., Geota A.E., Howard J.A.: The Observation of a Large Gauche Preference when 2-Fluoroethylamine and 2-Fluoroethanol Become Protonated. *Org. Biomol. Chem.* 2 (2004) 732-740
8. Banks J.W., Batsanov A.S., Howard J.A.K., O'Hagan D., Rzepa H.S., Martin-Santamaria S.: The Preferred Conformation of α -Fluoroamides. *J. Chem. Soc., Perkin Trans. 2.* 8 (1999) 2409-2411
9. Wiberg K.B., Murcko M. A., Laidig E.K., MacDougall P. J.: Origin of the "Gauche Effect" in Substituted Ethanes and Ethenes. *The Gauche Effect. J. Phys. Chem.* 96 (1992) 6956-6959 and references therein
10. Harris W.C., Holtzclaw J.R., Kalasinsky V.F.: Vibrational Spectra and Structure of 1,2-Difluoroethane: Gauche-Trans Conformers. *J. Chem. Phys.* 67 (1977) 3330-3338
11. Friesen D., Hedberg K.: Conformational Analysis. 7. 1,2-Difluoroethane. An Electron-Diffraction Investigation of the Molecular Structure, Composition, Trans-Gauche Energy and Entropy Differences, and Potential Hindering Internal Rotation. *J. Am. Chem. Soc.* 102 (1980) 3987-3994
12. Takeo H., Matsumura C., Morino Y.: Microwave Spectrum and Molecular Structure of Gauche-1,2-Difluoroethane. *J. Chem. Phys.* 84 (1986) 4205-4210
13. Martell J.M., Boyd R.J.: An Ab initio Study of the Series $C_2H_nF_{6-n}$ (n=0-6): Geometries, Total Energies, and C-C Bond Dissociation Energies. *J. Phys. Chem.* 96 (1992) 6287-6290

14. Durig J.R., Liu J., Little T.S., Kalasinsky V.F.: Conformational Analysis, Barriers to Internal Rotation, Vibrational Assignment, and Ab Initio Calculations of 1,2-Difluoroethane. *J. Phys. Chem.* 96 (1992) 8224-8233
15. Wiberg K.B., Keith T.A., Frisch M.J., Murcko M.: Solvent Effects on 1,2-Dihaloethane Gauche/Trans Ratios. *J. Phys. Chem.* 99 (1995) 9072-9079
16. Ramasami P.: Gauche and Trans Conformers of 1,2-Dihaloethanes: A Study by Ab Initio and Density Functional Theory Methods. *Lecture Series on Computer and Computational Sciences, Vol. 1, Brill Academic Publishers, The Netherlands (2005) 732-734*
17. Gaussian 03, Revision C.02, Frisch M.J., Trucks G.W., Schlegel H.B., Scuseria G.E., Robb M.A., Cheeseman J.R., Montgomery J.A., Jr., Vreven T., Kudin K.N., Burant J.C., Millam J.M., Iyengar S.S., Tomasi J., Barone V., Mennucci B., Cossi M., Scalmani G., Rega N., Petersson G.A., Nakatsuji H., Hada M., Ehara M., Toyota K., Fukuda R., Hasegawa J., Ishida M., Nakajima T., Honda Y., Kitao O., Nakai H., Klene M., Li X., Knox J.E., Hratchian H.P., Cross J.B., Bakken V., Adamo C., Jaramillo J., Gomperts R., Stratmann R.E., Yazyev O., Austin A.J., Cammi, R. Pomelli C., Ochterski J.W., Ayala P.Y., Morokuma K., Voth G.A., Salvador P., Dannenberg J.J., Zakrzewski V.G., Dapprich S., Daniels A.D., Strain M.C., Farkas O., Malick D.K., Rabuck A.D., Raghavachari K., Foresman J.B., Ortiz J.V., Cui Q., Baboul A.G., Clifford S., Cioslowski J., Stefanov B.B., Liu G., Liashenko A., Piskorz P., Komaromi I., Martin R.L., Fox D.J., Keith T., Al-Laham M.A., Peng C.Y., Nanayakkara A., Challacombe M., Gill P.M.W., Johnson B., Chen W., Wong M.W., Gonzalez C., Pople J.A.: *Gaussian, Inc., Wallingford CT, 2004.*
18. http://www.gaussian.com/gv_plat.htm
19. Hurley M.M., Pacios L.F., Christiansen P.A., Ross R.B., Ermler W.C.: Ab Initio Relativistic Effective Potentials with Spin-Orbit Operators. II. K through Kr. *J. Chem. Phys.* 84 (1986) 6840-6853
20. Durig J.R., Pan C., Witkowski W., Guirgis G.A.: Conformational Stability, Infrared and Raman spectra, and Vibrational assignment of Ethyl Bromogermane. *Can. J. Chem.* 82 (2004) 964-977
21. Tanabe K.: Calculation of Infrared Band Intensities and Determination of Energy Differences of Rotational Isomers of 1,2-Dichloro-, 1,2-Dibromo- and 1-Chloro-2-Bromoethane. *Spectrochim. Acta* 28 (1972) 407-424

Model Dependence of Solvent Separated Sodium Chloride Ion Pairs in Water-DMSO Mixtures

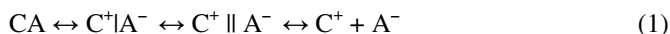
A. Asthana, A.K. Chowdhury, A.K. Das, and B.L. Tembe

Department of Chemistry, I.I.T. Bombay,
Powai, Mumbai 400 076, India

Abstract. Constrained molecular dynamics simulations have been used to investigate ion pairing in water - DMSO mixtures. The potentials of mean force between the sodium - chloride ion pair are constructed by estimating the mean forces between the ion pair at various interionic separations and then integrating the mean forces. Two compositions of the solvent mixture with DMSO mole fractions of 0.21 and 0.35 are considered. Two model potentials for water and DMSO have been considered. One of the main observations is that the contact ion pair which is dominantly present in both the individual solvents is conspicuously absent in the mixture compositions studied. While solvent separated ion pairs dominate in all the mixture compositions, there is a presence of a second solvent separated ion pair in the water-DMSO mixture of composition with mole fraction of DMSO = 0.21. The potentials of mean force are verified by dynamical trajectories of the ion pair. The dynamics of the solvation shells has also been investigated in detail.

1 Introduction

Ion pairs play a very important role in a variety of chemical and biochemical reactions. The solvation of ions and ion pairs influences the nature of reactions in solution media.¹⁻⁵ The concept of two ion pair states, namely, the contact ion pair (CIP) and the solvent separated ion pair (SSIP), was proposed independently by Winstein⁶ and by Fuoss.⁷ The general scheme for any solvolysis reaction or any reaction that involves participation of solvent can be represented by



where the $C^+|A^-$ represents the CIP and $C^+||A^-$ represents the SSIP. The two ion pair states are characterized by the existence of well-defined barriers in the ion-ion potential of mean force (PMF) derived through methods such as constrained MD simulations. These ion pairs also exist in mixed solvents and the transformation of the CIP into SSIP or vice versa involves the motion over an energy barrier between these two states.

The first attempt to perform model calculations of solvations of alkali halides in DMSO was made by Goldenberg et al.⁸ Energy calculations were carried out on a cluster of an ion with 4, 5 and 6 molecules of solvent and on ion pairs with 14 DMSO molecules respectively. Ten point charges and nine polarizable bonds represented a DMSO molecule. It was found that calculated ion solvation energies were in quantitative agreement with experimental data.

The first real MD simulation of liquid DMSO represented by 216 rigid molecules was performed by Rao and Singh.⁹ They determined relative differences in free energies of solvation between two different solutes, including Na⁺, K⁺, Cl⁻ and Br⁻ ions, in DMSO as well as in water and methanol. Their study showed that strong ion-dipole interaction, and a well-defined co-ordination sphere of five DMSO is formed around Na⁺ ion in liquid DMSO. Cl⁻ ion was found to be surrounded by about seven DMSO molecules and there is no well-defined co-ordination. A detailed investigation of the solvation phenomena of sodium and chloride ion pairs in liquid DMSO was undertaken by Madhusoodanan and Tembe.¹⁰ The potential mean force, the structure of solvation shells around the Na⁺ - Cl⁻ pair and the dynamics of association of Na⁺ - Cl⁻, Na⁺ - Na⁺ and Cl⁻ - Cl⁻ pairs were studied by using the method of constrained MD.¹¹⁻¹³

Neutron diffraction and MD simulation based on the data, generated from the neutron diffraction experiments, were performed by Luzar et al to study the structure of water-DMSO binary mixture.¹⁴ The results showed the DMSO oxygen to be strongly hydrogen bonded, but the methyl groups were surrounded by loose hydrogen bonded cages of water molecules. No evidence for hydrophobic association of DMSO was observed. Chalaris and Samios¹⁵ studied the dynamics of water-DMSO mixtures using different effective potential model for both water and DMSO. They used the SPC, SPC/E and TIPS2 models for water and the P2 model for DMSO. The SPC/E gave the good results for water self-diffusion coefficient at low DMSO concentrations.

One of the recent MD simulation study of DMSO-water mixtures containing either an ionic or a neutral atomic solute was performed by Chowdhury and Chandra.¹⁶ The diffusion coefficients of both ionic and neutral solutes was found out using SPC/E potential model for water and 4-site P2 model of Luzar and Chandler for representing the DMSO molecules¹⁴.

In the next section, we outline the methodology used in the present study. This is followed by the results and discussions and conclusions.

2 Methodology

Simulation studies have been carried out on systems consisting of sodium and chloride ions in water-DMSO mixtures. Positions and velocities are calculated using the Verlet algorithm with a sufficiently small time step so that the constraints of the rigidity of the solvent molecules are satisfied with little difficulty during the simulations.¹⁷ The simulation study considers both the ions in one cubic simulation box along with solvent molecules. To calculate the potentials of mean force between the ions, we first calculate the average force between the ions using

$$\mathbf{F}(\mathbf{r}) = \mathbf{F}^{\text{direct}}(\mathbf{r}) + \Delta\mathbf{F}(\mathbf{r}) \quad (2)$$

Where $\Delta\mathbf{F}(\mathbf{r})$ is the contribution to the mean force from the solvent and $\mathbf{F}^{\text{direct}}(\mathbf{r})$ is the bare ion-ion potential. The mean solvent force $\Delta\mathbf{F}(\mathbf{r})$ is obtained by ensemble averaging the instantaneous solvent force $\Delta\mathbf{F}(\mathbf{r},t)$ between the ions.

$$\Delta\mathbf{F}(\mathbf{r},t) = \mu \left[\mathbf{F}_{\text{Na}^+}(\mathbf{r},t) / m_{\text{Na}^+} - \mathbf{F}_{\text{Cl}^-}(\mathbf{r},t) / m_{\text{Cl}^-} \right] \cdot \hat{\mathbf{r}} \quad (3)$$

$$\Delta\mathbf{F}(\mathbf{r}) = \langle \Delta\mathbf{F}(\mathbf{r},t) \rangle \quad (4)$$

$\Delta F(r)$ is calculated at various fixed distances between the ions (in the range of 2.5 to 8 Å). In Eq. (2) \hat{r} is the unit vector between the sodium and the chloride ions. The masses of the ions are denoted by m_{Na^+} and m_{Cl^-} and μ is their reduced mass. Typically 20 to 50 ps of equilibration is followed by production runs of 100 ps.

The potential of mean force is obtained by integrating the mean force as

$$W(r) = - \int_{r_0}^r F(r) dr \quad (5)$$

We need a suitable value of $W(r)$ at r_0 to complete the integration. This is done by assuming that at a distance of $r_0 = 8 \text{ \AA}$, $W(r)$ is given by the formula

$$W(r) = q_1 q_2 / \epsilon r \quad (6)$$

Where ϵ is the bulk dielectric constant. Its value is taken from the experimental data at the appropriate solvent mole fraction. The data is given in Table 1.

Our results for the SPC model of water and the P2 model of DMSO have been already reported.¹⁸⁻¹⁹ In the present article, we have used the SPC/E model of water and the P1 model of DMSO. While the parameters of the SPC/E model of water are rather close to those of the SPC model, the P1 model of DMSO differs considerably from the P2 model in that the charge on S in P1 is nearly four times the charge of the P2 model and the methyl groups in P1 are uncharged. The energy parameter of methyl in P1 is also twice that of the P2 model. There are two reasons for choosing the new parameters. One is that the new model has been to give better values of the individual ionic diffusion coefficients. The other reason is to find out the model dependence of the second SSIP proposed earlier.

The interparticle potentials are the usual Lennard Jones plus the Coulomb terms. The site-site ion-solvent and solvent-solvent potential is represented as

$$U_{\alpha\beta}(r) = (A_{\alpha\beta}/r^{12}) - (C_{\alpha\beta}/r^6) + (q_{\alpha}q_{\beta}/r) \quad (7)$$

Where α and β denote a pair of interaction sites on different molecules, r is the site separation, q_{α} is a point charge located at site α . The terms $A_{\alpha\beta}$ and $C_{\alpha\beta}$ are given by

$$A_{\alpha\beta} = 4\epsilon_{\alpha\beta}(\sigma_{\alpha\beta})^{12} \quad (8)$$

$$C_{\alpha\beta} = 4\epsilon_{\alpha\beta}(\sigma_{\alpha\beta})^6 \quad (9)$$

Here, $\epsilon_{\alpha\beta}$ and $\sigma_{\alpha\beta}$ are the energy and the distance parameters in the Lennard-Jones potential. The cross interactions have been obtained by using the Lorentz-Berthelot rules

$$\sigma_{\alpha\beta} = \frac{1}{2} (\sigma_{\alpha\alpha} + \sigma_{\beta\beta}) \quad (10)$$

$$\epsilon_{\alpha\beta} = (\epsilon_{\alpha\alpha} \epsilon_{\beta\beta})^{1/2} \quad (11)$$

These parameters of the potentials used have been given in Table 1. Table 2 gives the details of the simulation cells.

Table 1. Site parameters for models of water and DMSO

Solvent	Site	ϵ	$\sigma / \text{\AA}$	q / e^*	ϵ	$\sigma / \text{\AA}$	q / e^*
		(10J/mol)			(10J/mol)		
Water	O	0.6052	3.156	-0.82	0.6052	3.166	-0.8476
	H	0.0	0.0	0.41	0.0	0.0	0.4238
DMSO	S	0.84403	3.56	0.139	0.99741	3.4	0.54
	O	0.27586	2.94	-0.459	0.29922	2.80	-0.54
	CH ₃	0.66852	3.60	1.60	1.23	3.80	0.0

* e is the magnitude of the electronic charge.

Table 2. Details of the two mixture compositions showing compositions, number of solvent molecules, box lengths, densities and the dielectric constants^{18,19}

Composition	x_1	x_2	N_1	N_2	$L (\text{\AA})$	$\rho (\text{kgL}^{-1})$	ϵ
1	0.21	0.79	53	197	22.784	1.0725	74.85
2	0.35	0.65	88	162	24.764	1.0927	69.97

3 Results and Discussion

The mean forces between the ions for the two compositions are shown in Figs 1 and 2 for compositions 1 and 2 respectively.

The direct force is attractive and the solvent contribution is repulsive in both the compositions. The total contribution lies between -10 to 5 ($\text{kT}/\text{\AA}$) in the distance range of 3.5 to 8 \AA . Both these forces are integrated from r_0 to r to get the potential of mean force (PMF). The asymptotic form of the PMF of the continuum model of the solvent mixture at $r = r_0$ is used during the integration. The results for the two mixtures are shown in Figs 3 and 4 respectively.

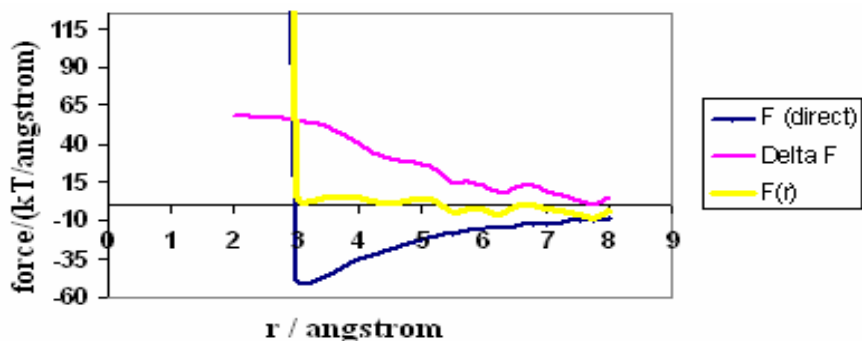


Fig. 1. The ion – ion force $F(r)$, the contribution due to the solvent $F(r)$ and the total force between the ion pair for composition 1 ($x_{\text{DMSO}} = 0.21$)

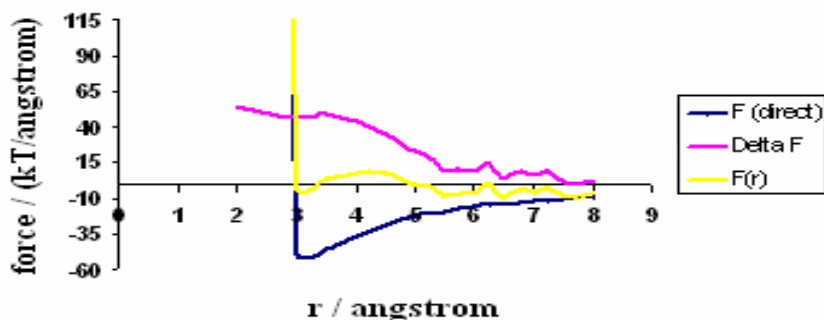


Fig. 2. The ion – ion force $F(r)$, the contribution due to the solvent $F(r)$ and the total force between the ion pair for composition 1 ($x_{\text{DMSO}} = 0.35$)

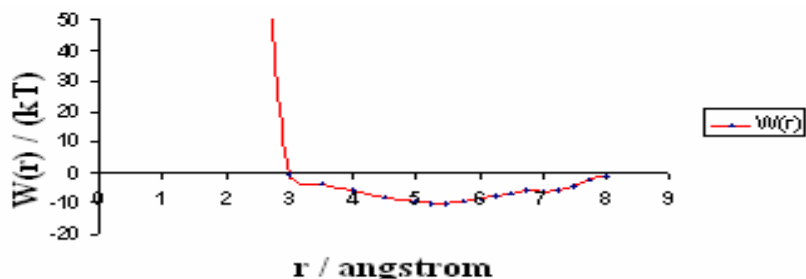


Fig. 3. Potential of mean force for the sodium chloride ion pair in composition 1 ($x_{\text{DMSO}} = 0.21$)

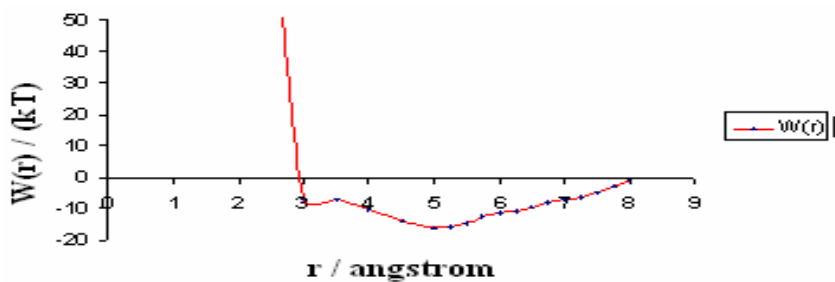


Fig. 4. Potential of mean force for the sodium chloride ion pair in composition 2 ($x_{\text{DMSO}} = 0.35$)

Both the PMFs show the absence of a short range minimum around 3 \AA corresponding to the contact ion pair and the presence of a strong solvent-separated ion pair (SSIP) in the range of 4.5 to 5.5 \AA with a well depth of around 10 kT . The first composition (21% DMSO) shows the presence of a second SSIP minimum near 7 \AA , although this minimum is not separated from the first SSIP minimum with a very

clear barrier. The verification of these PMFs was done by studying the ion pair trajectories which are initiated at various separations. Figs 5 and 6 show these ion pair trajectories for the mixtures.

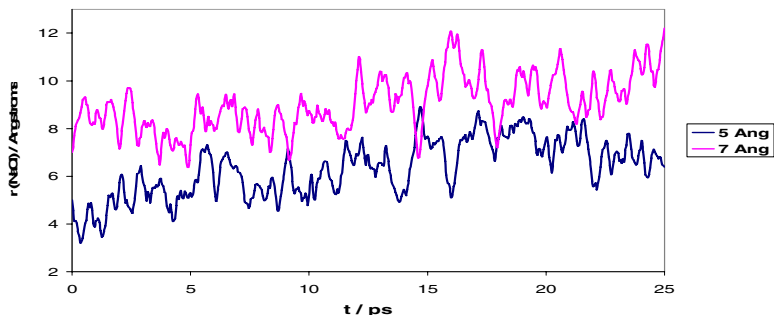


Fig. 5. Ion pair trajectories initiated at 5 and 7 Å respectively for mixture 2 ($x_{\text{DMSO}} = 0.21$)

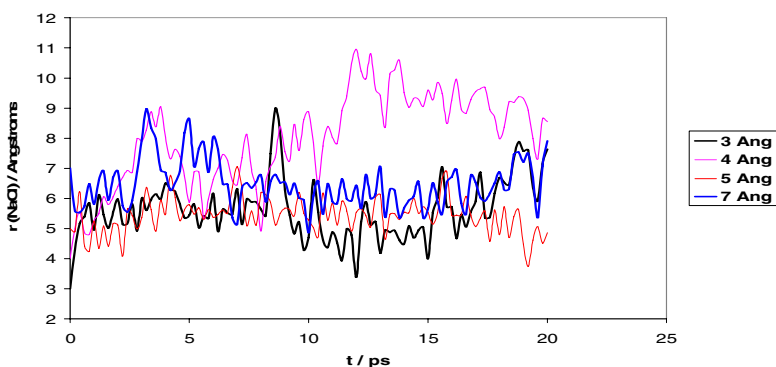


Fig. 6. Ion pair trajectories initiated at 3, 4, 5 and 7 Å for mixture 2 ($x_{\text{DMSO}} = 0.35$)

The trajectories are initiated with the ion pair separations of in the range of 3 to 7 Å. The trajectories initiated at small ion pair separations of 3 Å reach large ion separations (of greater 4 Å) within a few ps. The ion pair trajectories initiated at large ion pair separations diffuse very slowly indicating that the long distance ion pairs have a very significant stability in the solvent mixtures. One of the reasons for this dominance of the solvent separated ion pair is the strong interaction between water and DMSO and the competition of both water and DMSO molecules to get into the first solvation shell of the ion pair.²⁰ There appears to be a disruption of the contact ion pair formed in the pure solvents by the intrusion of the other solvent molecules in the first shell. The spread out solvent separated ion pair in the distance range of 4 to 6 Å implies that there is a significant flux of molecules from the first shell to the bulk solvent and vice versa and this has been noticed in the simulations. The diffusivity of

the trajectories in the present case²¹ is similar to our previous work^{18, 19} and we are investigating these aspects in greater detail.

4 Conclusions

The present results for the PMFs have a close similarity to the findings of the earlier model calculations.¹⁸ There is a striking absence of contact ions pairs and a dominance of solvent separated ions pairs. These are confirmed by detailed dynamical trajectories. While the parameters of the SPC/E model of water are rather close to those of the SPC model, the P1 model of DMSO differs considerably from the P2 model with respect to the charge on S and the size of the methyl groups. We are investigating the details of this clustering as well as the long time trajectories of the ion pairs. We may find larger deviations in the PMFs between the two models for compositions wherein the mole fractions of DMSO are larger. These studies are under way.

References

1. Hubbard, J., Wolynes, P.: *The Chemical Physics of Solvation, Part C*: (Ed) Dogonadze, R.R., Kalman, E., Kornyshev, A. A., Ulstrup, J., Elsevier, New York (1988)
2. Buchner, R.: *Dielectric Spectroscopy of Solutions*, in: (eds) Samios, J., Durov, V.A. *Novel Approaches to the Structure and Dynamics of Liquids: Experiments, Theories and Simulations*, NATO Science Ser. II: Mathematics, Physics and Chemistry, Vol. 133, pp. 265-288, Kluwer, Dordrecht (2004)
3. Bester-Rogac, M., Neueder, R., Barthel, J.: *J. Solution Chem.* 28 (1999) 1077-1092
4. Tongraar, A., Rode, B. M.: *Phys.Chem.ChemPhys.*, 6 (2004) 411-416
5. Schwenk, C. F., Hofer, T. S., Rode, B. M.: *J. Phys. Chem.A* 108 (2004) 1509-1514
6. Winstein, S., Clippinger, E., Fainber, A. H., Robinson, G.C.: *J. Am. Chem. Soc.* 26 (1954) 1696.
7. Sadek, H., Fouuss, R.: *J. Am. Chem. Soc.*, 76 (1954) 5897
8. Goldenberg, M.S., Kruus, P., Luk, S. F. K.: *Can. J. Chem.* 53 (1975) 1007
9. Rao, B. G., Singh, U. C.: *J. Am. Chem.Soc.* 112 (1990) 3803
10. Madhusoodanan, M.; Tembe, B.L.: *J. Phys.Chem.* 98 (1994) 7090
11. Guardia, E., Rey, R., Padro, J. A.: *Chem. Phys.* 155 (1991) 187
12. Sesse, G.;Guardia, E., Padro, J. A.: *J. Chem. Phys.* 99 (1995) 12647
13. Das, A. K., Madhusoodanan, M., Tembe, B. L.: *J. Phys. Chem. A*101 (1997), 2862
14. Luzar, A., Chandler, D.: *J. Chem. Phys* 98 (1993) 8160
15. Chalaris, M., Samois, J.: *J. Mol. Liq.*, 98 (2002) 399
16. Chowdhury, S.; Chandra, A.: *J. Chem. Phys.* 119 (2003) 4360
17. Allen, M. P., Tildesley D. J.: *Computer Simulation of Liquids*, Oxford University Press (1987)
18. Das, A. K., Tembe, B. L.: *J. Chem. Phys.* 111 (1999) 7526-7536
19. Das, A. K.: *Simulation Studies on the Sodium Chloride Ion Pair in DMSO and Water-DMSO Mixtures*, Ph.D Thesis, IIT Bombay (1998)
20. Das, U.: "Molecular dynamics study of the solvation of ions in water-DMSO Mixtures", M.Sc. Thesis, Department of Chemistry, IIT Bombay, 2003
21. Asthana, A.: "Dynamics of ions and molecules in solvent mixtures like Water-DMSO", M.Sc. Thesis, Department of Chemistry, IIT Bombay, 2005

Fault Distinguishability of Discrete Event Systems

Iwan Tabakow

Institute of Applied Informatics, Wrocław University of Technology, Poland
iwan.tabakow@pwr.wroc.pl

Abstract. The subject of this paper is the theory of fault distinguishable discrete event systems. Any such system is modelled by a live, bounded, and reversible place-transition net. The notions of D-partition of the set of places P of a given place-transition net N and net k -distinguishability are first introduced. The system k -distinguishability measure is obtained in a unique way from the place-invariant matrix. For a large value of k , the system model is extended by using some set of additional places called test points. It is shown that the test point placement process will not change the above-assumed original net properties. Several examples are given.

1 Introduction

The use of Petri net models in diagnosis and reliable design of event-driven systems is a subject of interest to researchers since more than twenty years. In general, the most of the studies in this area focus attention on dynamical analysis concerning specification and implementation of some fault detection, fault diagnosis and/or fault recovery procedures, e.g. using partially stochastic Petri nets [1], or also using trace analysis [5], etc. The study of the system fault indistinguishability properties seems to be important because of the following two reasons. First, we have an additional possibility of describing the critical components of the considered system. Second, there exists a possibility of using some simple and at the same time exact tools for improving the system (self-) diagnosis capabilities in the early stages of its design [2,8,9,10,11,12].

The subject of this paper is the theory of fault distinguishable discrete event systems. Any such system is modelled by a live, bounded, and reversible place-transition net. The notions of D-partition of the set of places P of a given place-transition net N and net k -distinguishability are first introduced. Next these two notions are extended to the set of all vertices, i.e. places and transitions of N . So the problem of fault identification of the vertices of N is transformed as a problem of fault identification of the places of a new net N' called a net simulator of N . Any transition in N' is assumed to be fault-free. Then the corresponding net place invariants are computed. The system k -distinguishability measure is obtained in a unique way from the place-invariant matrix. For a large value of k , the system model is extended by using some set of additional places called test points. It is shown that the test point placement process will not change the above-assumed original net properties. Several examples are given.

2 Basic Notions

In general any place-transition net $N =_{df} (T, P, A, M_0, K, W)$, where (T, P, A) is a finite net containing sets of *transitions*, *places*, and *arcs* called also *edges*, $K : P \rightarrow (IN_\omega - \{0\})$ and $W : A \rightarrow \mathbb{N}$ are the corresponding *place capacity* and *edge multiplicity* (called also *weight*) functions, respectively. The *initial marking vector* $M_0 : P \rightarrow IN_\omega$, where \mathbb{N} denotes the set of all natural numbers, $IN =_{df} \mathbb{N} \cup \{0\}$, $IN_\omega =_{df} IN \cup \{\omega\}$, and ω is an infinite number such that: $\omega + k = \omega$ and $k < \omega$ (for any $k \in IN$) [6,7]. The *forward marking class* of N , i.e. $[M_0 > =_{df} \{ M \in IN_\omega^P / \exists \tau \in T^* (M_0[\tau > M) \}$.

In the next considerations we shall assume N is a pure, live and bounded net. In the case of manufacturing systems the *net reversibility property* is also required. The net *P-invariants* are computed using $\underline{N} \cdot \underline{i} = \underline{0}$, where \underline{N} is the *PN-connectivity matrix* of N . The *support* of any P-invariant \underline{i} with respect to N (in short: wrt N) is defined as follows: $\text{supp}(\underline{i}) =_{df} \{ p \in P / \underline{i}(p) \neq 0 \} \subseteq P$. Let \mathcal{J} be the set of all (positive) P-invariants of N and $\mathcal{J}' \subseteq \mathcal{J}$ is a subset. The *P-invariant matrix* of N wrt \mathcal{J} is introduced as follows: $\underline{J} : \mathcal{J} \times P \rightarrow IN$, where $\underline{J}(\underline{i}, p) =_{df} \underline{i}(p) \in IN$. For convenience only, we shall assume below that the P-cover \mathcal{J} of N is a set of all positive and minimal P-invariants. Also we shall use the notion of the *revised P-invariant matrix* of N , defined as: $\underline{\rho} : \mathcal{J} \times P \rightarrow \{0,1\}$, where $\underline{\rho}(\underline{i}, p) =_{df} 1$ iff $\underline{i}(p) \neq 0$ [2]. For simplicity, it is assumed below N have a P-cover. Otherwise, this method is also applicable. In the last case some additional test points is necessary to be introduced.

3 Net k-Distinguishability and Test Points

Let $[M_0 >_\alpha =_{df} [M_0 > \cup \{ M_\alpha \}$, where M_0 is the initial marking and M_α is a marking of N such that $M_\alpha \notin [M_0 >$. We shall say M_α is a *faulty marking*. Since $M \underline{i} = M_0 \underline{i}$ (for any $M \in [M_0 >$ and $\underline{i} \in \mathcal{J}'$) [6] then $\Delta M \underline{i} = 0$, where $\Delta M =_{df} M - M_0$. The last property is satisfied for any P-invariant $\underline{i} \in \mathcal{J}'$. Hence we can obtain $\underline{J} \Delta M^T = \underline{0}$. Therefore for $M \in [M_0 >_\alpha$ the above equation may be violated. Thus we have: $\underline{J} \Delta M^T = \underline{a} \in \{0,1\}^{|\mathcal{J}'|}$ (for any $M \in [M_0 >_\alpha$, obviously $\underline{a} = \underline{0}$ iff $M \in [M_0 >$). Without losing any generality, below $(\underline{a})_s \neq 0$ are interpreted as $(\underline{a})_s = 1$ ($s \in \{1, \dots, |\mathcal{J}'|\}$). Hence, in accordance with [4], any $(\underline{a})_s = 1$ will correspond to some subset of places $\text{supp}(\underline{i}_s) \subseteq P$ having a (potentially) faulty behaviour. Let $P \supseteq \Omega(\underline{a}) =_{df} \bigcap \text{supp}(\underline{i}_s) / (\underline{a})_s = 1 \cap \bigcap \text{supp}(\underline{i}_s)' / (\underline{a})_s = 0$, where $\text{supp}(\underline{i}_s)' =_{df} P - \text{supp}(\underline{i}_s)$ is the corresponding set complement operation. So, like [3] the notion of D-partition can be introduced. Below are used some basic notions given in [8].

Definition 1

By a *D-partition of the set of places* P of a given place-transition net N wrt the P-cover \mathcal{J} of N , denoted by $\Omega(N, \mathcal{J})$, or Ω if N and \mathcal{J} are understood, we shall mean the (multi) family $\Omega =_{df} \{ \Omega(\underline{a}) / \underline{a} \in \{0,1\}^{|\mathcal{J}'|} \}$.

Proposition 1 [8]

- (a) $\Omega(\underline{0}) = \emptyset$,
- (b) $\forall \underline{a}, \underline{b} \neq \underline{0} (\underline{a} \neq \underline{b} \Rightarrow \Omega(\underline{a}) \cap \Omega(\underline{b}) = \emptyset)$, and
- (c) $\cup \Omega(\underline{a}) / \underline{a} \in \{0,1\}^{|\mathcal{J}|} = P$. □

The notion of a k -distinguishable place-transition net under a D -partition of the set of places P of N is given in the next definition.

Definition 2

The Petri net N is a k -distinguishable net under Ω iff

- (i) $\exists \Omega(\underline{a}) \in \Omega (|\Omega(\underline{a})| = k)$ and
- (ii) $\forall \Omega(\underline{a}) \in \Omega (|\Omega(\underline{a})| \leq k)$.

The *support* of any D -partition is defined as follows: $\text{supp}(\Omega) =_{\text{df}} \{ \Omega(\underline{a}) \in \Omega / \Omega(\underline{a}) \neq \emptyset \}$. Let $\pi(P)$ be the partition generated by the set of subsets of places (i.e. classes), such that each class consists of places having identical columns in the revised P -invariant matrix ρ of N . The following proposition is satisfied (a more formal proof is given in [11]).

Proposition 2 [2]

$$\text{supp}(\Omega) = \pi(P). \quad \square$$

Definition 3

Let N be a place-transition net. Then N is a k -distinguishable net iff $\exists N'$ (N' is a net simulator of N and N' is a k -distinguishable net under Ω').

Definition 4

Let $p_{k_0} \in P$ be a given place of N such that the pre-set $\bullet p_{k_0} =_{\text{df}} \{t_1\}$ and the post-set $p_{k_0} \bullet =_{\text{df}} \{t_2\}$, where t_1 and t_2 are two different transitions of N . The additional place $p_{k_0}' \in \bullet t_1 \cap t_2 \bullet$ is said to be a *test point associated with* p_{k_0} iff the initial marking \hat{M}_0 of the obtained net \hat{N} is specified as follows: $\hat{M}_0(p) =_{\text{df}}$ if $p = p_{k_0}'$ then $\max\{M(p_{k_0}) / M \in [M_0 >] - M_0(p_{k_0})\}$ else $M_0(p)$ fi (for any $p \in \hat{P} =_{\text{df}} P \cup \{p_{k_0}'\}$).

It can be observed that in some cases the considered Petri net may be *maximally indistinguishable*, e.g. a net which is a state-machine net and a marked graph at the same time. Then the corresponding P -cover will contain only one P -invariant having all components equal to one.

Proposition 3

Let N be a directed elementary cycle having m places ($m > 1$). Then N becomes $(2m - r)$ -distinguishable if r additional test points are placed ($1 \leq r \leq 2m - 1$). {Df.3, Prop.2}□

A generalisation of Definition 3 for *non-ordinary place-transition nets* (i.e. nets having some edges $a \in A$ with weights $w(a) \neq 1$) is omitted here. Any such generalisation of the last definition for place-transition nets, which are not ordinary, would require an isomorphism between the corresponding reachability graphs $RG(N)$ and $RG(\hat{N})$ (see Example 1 and Theorem 2 given below). Let $P_{\underline{a}} =_{df} \Omega(\underline{a})$ (for any $\underline{a} \in \{0,1\}^{|J|}$). Obviously $P_{\underline{a}} \in \text{supp}(\Omega)$ if $P_{\underline{a}} \neq \emptyset$.

Definition 5

Let $P_{\underline{a}} \neq \emptyset$, $T_{\underline{a}} =_{df} \bullet P_{\underline{a}} \cup P_{\underline{a}} \bullet$ and $A_{\underline{a}} =_{df} A \cap ((T_{\underline{a}} \times P_{\underline{a}}) \cup (P_{\underline{a}} \times T_{\underline{a}}))$. The corresponding subnet $N_{\underline{a}} =_{df} (T_{\underline{a}}, P_{\underline{a}}, A_{\underline{a}})$ of N is called a *graphical representation* of $P_{\underline{a}}$. We shall say $N_{\underline{a}}$ is a *marked graph component* (or *MG-component*) iff $\forall p \in P_{\underline{a}} (|\bullet p| = |p \bullet| = 1)$. The subset of places $P_{\underline{a}}$ is said to be a *MG-component generator*.

Theorem 1[10]

Assume that N a live and bounded place-transition net having $|P| \geq 2$ and $\text{supp}(\Omega) =_{df} \{P_{\underline{a}_1}, P_{\underline{a}_2}, \dots, P_{\underline{a}_n}\}$, where $1 \leq n < |P|$. If any $P_{\underline{a}} \in \text{supp}(\Omega)$ is a MG-component generator then N can be transformed into a 1-distinguishable net by using $(|P| - n)$ test points. □

It is obvious that any P-invariant in N can be extended as a P-invariant in \hat{N} by assuming 0's relating to the corresponding test point components, i.e. the following proposition is satisfied.

Proposition 4

If \underline{i} is a P-invariant in N then $\hat{\underline{i}} =_{df} (\underline{i}, \underline{0})$ is a P-invariant in \hat{N} , where the vector size of $\underline{0}$ is related to the number of used test points. □

Proposition 5

If p_{k_0}' is a test point associated with $p_{k_0} \in P$ in N then $\hat{\underline{i}}_{k_0}$ is a P-invariant in \hat{N} , where $\text{supp}(\hat{\underline{i}}_{k_0}) = \{p_{k_0}, p_{k_0}'\}$.

Proof

Assume that p_{k_0}' is a test point associated with $p_{k_0} \in P$ in N . According to Definition 4 $p_{k_0}' (p_{k_0})$ is at the same time an input (output) place to t_1 and an output (input) place to t_2 . By definition, a vector \underline{x} is a P-invariant iff $\hat{N} \cdot \underline{x} = \underline{0}$. Hence iff $\hat{\underline{i}} \cdot \underline{x} = \underline{0}$ (for any row-vector $\hat{\underline{i}}$ of \hat{N}). And so, there exist exactly two equations related to $t_1 (t_2)$ of the following form: $\dots - x_{p_{k_0}'} (+ x_{p_{k_0}'}) \dots + x_{p_{k_0}} (- x_{p_{k_0}}) \dots = 0$. The P-invariant $\hat{\underline{i}}_{k_0}$ is obtained by assuming $x_{p_{k_0}} = x_{p_{k_0}'} = 1$ and $x_i = 0$ (for any $x_i \neq x_{p_{k_0}}, x_{p_{k_0}'}$). □

Proposition 6

If \mathcal{J} is a P-cover of N then $\hat{\mathcal{J}} =_{df} \mathcal{J} \cup \{\hat{\mathbf{i}}_{k_0}\}$ is a P-cover of \hat{N} . {Prop.5} \square

In a natural manner, the last two propositions can be extended for non-ordinary place-transition nets. This is illustrated in the next example.

Example 1

Consider the hypothetical fragment shown in Figure 1(a) below. Let p' be a test point associated with $p \in P$ in N and $\hat{\mathbf{i}}$ be a P-invariant in \hat{N} such that $\text{supp}(\hat{\mathbf{i}}) =_{df} \{p, p'\}$. Using $\hat{N} \cdot \hat{\mathbf{i}} = \mathbf{0}$ (assuming $\hat{\mathbf{i}}(q) =_{df} 0$, for $q \in P - \text{supp}(\hat{\mathbf{i}})$) the following two equations can be obtained:

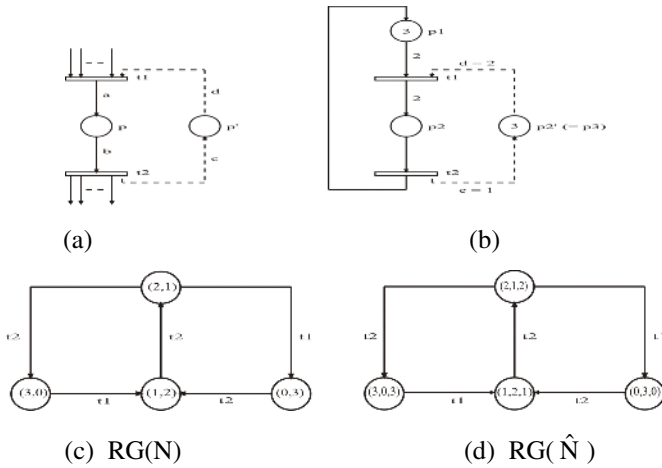


Fig. 1. A hypothetical fragment of non-ordinary place-transition net (a), an example test point placement (b), and the corresponding reachability graphs $RG(N)$ and $RG(\hat{N})$ (c and d, respectively)

$$\begin{aligned}
 a \cdot \hat{\mathbf{i}}(p) & - d \cdot \hat{\mathbf{i}}(p') = 0 \\
 -b \cdot \hat{\mathbf{i}}(p) & + c \cdot \hat{\mathbf{i}}(p') = 0
 \end{aligned}$$

Since the edge multiplicities a and b of N are a priori given then d and c can be defined in a unique way by assuming $\hat{\mathbf{i}}(p) = \hat{\mathbf{i}}(p') = 1$. Hence: $d =_{df} a$ and $c =_{df} b$. The obtained P-invariant $\hat{\mathbf{i}}$ is minimal and positive.

An example test point placement is shown in Figure 1(b) where an example live, bounded, and reversible place-transition net is presented. Assume that $T = \{t_1, t_2\}$ is a fault-free. According to Proposition 2 the considered net N is 2-distinguishable wrt the P-invariant $\hat{\mathbf{i}} = (1,1)$ having two identical columns. Let $p_3 =_{df} p_2'$ be a test point such that $\hat{M}_0(p_3) =_{df} \max\{M(p_2) / M \in [M_0 >] - M_0(p_2)\} = 3 - 0 = 3$.

According to Proposition 4 $\hat{\mathbf{i}} =_{df} (1,1,0)$ is a P-invariant in \hat{N} . Using $\hat{N} \cdot \underline{x} = \underline{0}$ the following two equations can be obtained:

$$\begin{aligned} -2 \cdot x_1 + 2 \cdot x_2 - d \cdot x_3 &= 0 \\ x_1 - x_2 + c \cdot x_3 &= 0 \end{aligned}$$

Let $x_1 = 0, x_2 = x_3 = 1$. Then $d = 2$ and $c = 1$. In accordance with Proposition 6 $\hat{\mathbf{i}}_2 =_{df} (0,1,1)$ is another P-invariant where $\text{supp}(\hat{\mathbf{i}}_2) = \{p_2, p_3\}$. In fact, we have $\hat{M}(p_2) + \hat{M}(p_3) = 3$ (for any $\hat{M} \in [\hat{M}_0 > \text{in } \hat{N})$. The obtained P-invariant matrix $\hat{\mathcal{J}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ has all columns different and \hat{N} is 1-distinguishable.

The corresponding reachability graphs $\text{RG}(N)$ and $\text{RG}(\hat{N})$ are shown in the above Figure 1(c) and (d), respectively. It can be observed that any M of N is a prefix of the corresponding \hat{M} of \hat{N} and the last two reachability graphs are isomorphic. Hence, the original boundedness, liveness, and reversibility properties of N are preserved in \hat{N} . □

Example 2

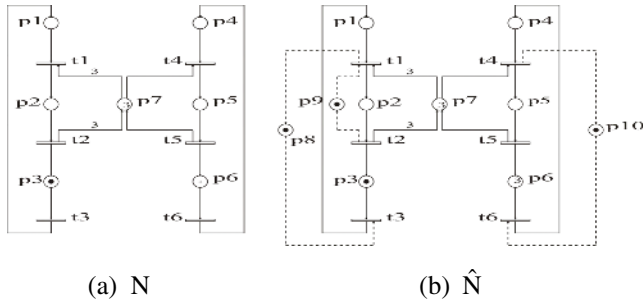


Fig. 2. A system consisting of one write- and three read-authorized processes (a) and a net distinguishability improving using test points p_8, p_9 and p_{10} (b)

Consider N of Figure 2(a) describing the behaviour of a system consisting of one write- and three read-authorized processes [4,6,7]. The following P-cover can be obtained: $\mathcal{J} = \{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\}$, where: $\hat{\mathbf{i}}_1 = (1,1,1,0,0,0,0)$, $\hat{\mathbf{i}}_2 = (0,0,0,1,1,1,0)$ and $\hat{\mathbf{i}}_3 = (1,4,1,0,1,0,1)$. According to Proposition 2, N is 3-distinguishable. The obtained test point improving is shown in Figure 2(b). The net N becomes 2-distinguishable for $\mathcal{J} = \{\hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3, \hat{\mathbf{i}}_4\}$ or also $\mathcal{J} = \{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_4\}$, where $\hat{\mathbf{i}}_4 = (0,3,0,0,1,0,1)$. In the last case the number of test points can be reduced to 2 (e.g. by removing p_9). {Df.2, Prop.2, T1} □

Theorem 2

Let N be live, bounded, and reversible place-transition net and p_{k_0} be a test point associated with p_{k_0} . Then \hat{N} is also live, bounded, and reversible.

Proof

Without losing any generality, assume that \mathcal{J} is a P-cover of N . Then $\hat{\mathcal{J}} \stackrel{\text{df}}{=} \mathcal{J} \cup \{\hat{i}_{k_0}\}$ is a P-cover of \hat{N} . Otherwise, a P-cover of N can be obtained by assuming additional test points. According to Definition 4 \hat{M}_0 is bounded. Hence \hat{N} is bounded.

Let $\bar{M} \stackrel{\text{df}}{=} \max\{M(p_{k_0}) / M \in [M_0 >]\}$ and $T(M) \stackrel{\text{df}}{=} \{t \in T / t \text{ is } M\text{-enabled in } N\}$. Assume that $t_1 \in T(M)$. Hence $t_1 \in T(\hat{M})$ iff $\hat{M}(p_{k_0}) + a \leq \bar{M}$ and $\hat{M}(p_{k_0}') \geq a$ (see the above Figure 1(a) assuming $p \stackrel{\text{df}}{=} p_{k_0}$ and $p' \stackrel{\text{df}}{=} p_{k_0}'$). However, in accordance with Definition 4 $\hat{M}(p_{k_0}) = M(p_{k_0})$ (for any $M \in [M_0 >]$). Hence $\hat{M}(p_{k_0}) + a = M(p_{k_0}) + a \leq \bar{M}$. Moreover, \hat{i}_{k_0} is a Boolean vector. Then $\hat{M}(p_{k_0}) + \hat{M}(p_{k_0}') = \bar{M}$ (for any $\hat{M} \in [\hat{M}_0 >]$ in \hat{N}). Hence: $a + \hat{M}(p_{k_0}) + \hat{M}(p_{k_0}') = \bar{M} + a$. Since $a + \hat{M}(p_{k_0}) \leq \bar{M}$ then $\bar{M} + \hat{M}(p_{k_0}') \geq \bar{M} + a$. Hence $\hat{M}(p_{k_0}') \geq a$ and $t_1 \in T(\hat{M})$.

Assume now that $t_2 \in T(M)$. Hence $t_2 \in T(\hat{M})$ iff $\hat{M}(p_{k_0}) \geq b$ and $\hat{M}(p_{k_0}') + b \leq \bar{M}$. Since $\hat{M}(p_{k_0}) = M(p_{k_0})$ the first condition $\hat{M}(p_{k_0}) \geq b$ is satisfied. Hence, using $\hat{M}(p_{k_0}) + \hat{M}(p_{k_0}') = \bar{M}$ we can obtain: $b + \hat{M}(p_{k_0}') \leq \bar{M}$.

Hence: $t \in T(M)$ iff $t \in T(\hat{M})$ (for $t \in \{t_1, t_2\}$). And so, the liveness and reversibility properties of N are preserved in \hat{N} . {Df.4, Prop.5, Prop.6} □

Test points can be placed independently each other. Hence Theorem 2 can be generalised for any finite subset of such points.

5 Conclusions

The above-considered approach gives a possibility of fault isolation in concurrent systems. This process is realised by using the Petri net model of the considered system. The degree of accuracy to which faults can be located, i.e. the diagnostic resolution is given in unique way by the obtained k -distinguishability measure. The complexity of the proposed method depends on the efficiency of the existing algorithms for computation of the P-cover, i.e. the set of P-invariants covering N . The choice of diagnosis strategies, i.e. combinational or also sequential is depending on the used time requirements for testing. Moreover, an additional cost-minimisation can be obtained by assuming the considered test point set as a “hardcore”. This approach

can be extended for higher level Petri nets, e.g. such as coloured nets or also to design self-diagnosable circuit realisations of Boolean interpreted Petri nets.

References

1. Aghasaryan A., Fabre E., Benveniste A., Boubour R. and Jard C., *Fault detection and diagnosis in distributed systems : an approach by partially stochastic Petri nets*. Discrete Event Dynamic Systems 8, 2 (Special issue on Hybrid Systems), June 1998, pages 203-231.
2. Immanuel B. and Rangarajan K., *System diagnosis and k-distinguishability in Petri nets*. Private communication, India (2001), 14pp.
3. Mayeda W., *Graph Theory*. John Wiley & Sons, Inc., New York (1972) 523 – 557.
4. Murata T., *Petri nets and their applications*. Journal Soc. Instrum. Control Eng. 22, Japan (1983) 6572.
5. Pietschker A. and Ulrich A., *A light-weight method for trace analysis to support fault diagnosis in concurrent systems*. Journal of Systemics, Cybernetics and Informatics, vol.1 no.6 (2003) 6pp.
6. Reisig W., *Petri Nets. An Introduction*. Springer-Verlag (1985) 15, 62 – 66.
7. Reisig W., *A Primer in Petri Net Design*. Springer-Verlag (1992) 25 – 33.
8. Tabakow I.G., *Using Petri net invariants in system diagnosis*. Petri Net Newsletter 58, Germany (2000) 21 – 31.
9. Tabakow I.G., *An introduction to the place-transition nets k-distinguishability*. Concurrency, Specification and Programming. Workshop. vol.2, Humboldt-Universität zu Berlin, Germany (2002), 355 – 369.
10. Tabakow I.G., *Using Test Points to Improve the Place –Transition Net k-Distinguishability*. Proc. of the 7th World Multiconference on Systemics, Cybernetics and Informatics SCI 2003, Orlando, Florida USA, July 27-30, vol. IX: Computer Science and Engineering II (2003), 173 - 178.
11. Tabakow I.G., *Using place invariants to isolate faults in concurrent systems*. Petri Net Newsletter 68, Germany (2005) 10 – 20.
12. Tabakow I.G., *Fault Diagnosis of Discrete Event Systems Using Place Invariants*. Ninth International Conference on Knowledge-Based & Intelligent Information & Engineering Systems KES'2005, Invited Session on Communicative Intelligence, Melbourne, Australia, September 14 - 16 (2005) in: LNCS, Springer-Verlag vol. 3682 (2005) 541 - 547.
13. Zhou M.C. and DiCesare F., *Petri net synthesis for discrete event control of manufacturing systems*. Kluwer Academic, Boston (1993) 233pp.

Modelling, Analyzing and Control of Interactions Among Agents in MAS*

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovak Republic
Frantisek.Capkovic@savba.sk, utrrcapk@savba.sk
<http://www.ui.sav.sk/home/capkovic/capkhome.htm>

Abstract. An alternative approach to modelling and analysis of interactions among agents in multiagent systems (MAS) and their control is presented in analytical terms. The reachability graph of the Petri net (PN)-based model of MAS is found as well as the space of feasible states. Trajectories representing the interaction processes among agents in MAS are computed by means of the mutual intersection of both the straight-lined reachability tree (from a given initial state towards the terminal one) and the backtracking reachability tree (from the desired terminal state towards the initial one however, oriented towards the terminal state). Control interferences are obtained on base of the most suitable trajectory chosen from the set of feasible ones.

1 Introduction

MAS are used in intelligent control, especially for a cooperative problem-solving [14]. To analyze complicated interactions among agents modelling of them is often used. The negotiation belongs to the most important interactions. It is the process of multilateral bargaining for mutual profit. In other words [13], [1], the negotiation is a decision process where two or more participants make individual decisions and interact with each other in order to reach a compromise. In [9] PN are used for e-negotiations activities. PN were chosen to model MAS too [12], [11]. On the base of previous experience [3], [4], [7] with PN-based modelling and control synthesis of the discrete event dynamic systems (DEDS) and the agent cooperation [5], [6] a new approach to modelling, analysis and control of the negotiation process is proposed here. The negotiation process is understood to be DEDS. The approach consists of: 1. creating the PN-based mathematical model of the negotiation process; 2. generating the space of feasible states which are reachable from the given initial state; 3. utilizing the reachability graph (RG) in order to find the feasible state trajectories to a prescribed feasible terminal state. After a thorough analyzing the set of possibilities, the most suitable strategy (the control trajectory) can be chosen.

* Partially supported by the Slovak Grant Agency for Science (VEGA) under grant # 2/6102/26.

2 The PN-Based Mathematical Model of DEDES

The analytical model of DEDES has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k \quad , \quad k = 0, \dots, N \quad (1)$$

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} \quad (2)$$

$$\mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k \quad (3)$$

where k is the discrete step of the dynamics development; $\mathbf{x}_k = (\sigma_{p_1}^k, \dots, \sigma_{p_n}^k)^T$ is the n -dimensional state vector of DEDES in the step k ; $\sigma_{p_i}^k \in \{0, 1, \dots, c_{p_i}\}$, $i = 1, \dots, n$ express the states of the DEDES elementary subprocesses or operations by 0 (passivity) or by $0 < \sigma_{p_i} \leq c_{p_i}$ (activity); c_{p_i} is the capacity of the DEDES subprocess p_i as to its activities; $\mathbf{u}_k = (\gamma_{t_1}^k, \dots, \gamma_{t_m}^k)^T$ is the m -dimensional control vector of the system in the step k ; its components $\gamma_{t_j}^k \in \{0, 1\}$, $j = 1, \dots, m$ represent occurring of the DEDES elementary discrete events (e.g. starting or ending the elementary subprocesses or their activities, failures, etc.) by 1 (presence of the corresponding discrete event) or by 0 (absence of the event); \mathbf{B} , \mathbf{F} , \mathbf{G} are structural matrices of constant elements; $\mathbf{F} = \{f_{ij}\}_{n \times m}$, $f_{ij} \in \{0, M_{f_{ij}}\}$, $i = 1, \dots, n$, $j = 1, \dots, m$ expresses the causal relations among the states of the DEDES (in the role of causes) and the discrete events occurring during the DEDES operation (in the role of consequences) by 0 (nonexistence of the corresponding relation) or by $M_{f_{ij}} > 0$ (existence and multiplicity of the relation); $\mathbf{G} = \{g_{ij}\}_{m \times n}$, $g_{ij} \in \{0, M_{g_{ij}}\}$, $i = 1, \dots, m$, $j = 1, \dots, n$ expresses very analogically the causal relations among the discrete events (causes) and the DEDES states (consequences); the structural matrix \mathbf{B} is given by means of the arcs incidence matrices \mathbf{F} and \mathbf{G} according to (2); $(\cdot)^T$ symbolizes the matrix or vector transposition. PN marking was denoted here by the letter \mathbf{x} usually denoting the state in system theory. Fuzzy PN [10] can be modelled analogically - see [2]. Higher-level PN models usually cannot be expressed in such a pure analytical form although they are advantageous as to the smaller dimensionality.

2.1 The PN-Based Model of the Negotiation Process Principle

The negotiation process itself consists of several principle activities [9]. Especially, the following ones are most important: defining the negotiation environment, initial contact of agents, offer(s) and counter offer(s) among them, evaluation of proposals, and outcomes of the negotiation process. The coordination plan of the negotiation process can be formally described by DEDES modelled by PN as it can be seen on the left in Fig. 1. The PN places represent the activities and the PN transitions represent the discrete events. The interpretation of the places and transitions is the following: $p_1 = \text{start}$; $p_2 = \text{define negotiation environment}$; $p_3 = \text{initial contact}$; $p_4 = \text{offer(s) and counter offer(s)}$; $p_5 = \text{evaluation}$; $p_6 = \text{outcomes}$; $p_7 = \text{end}$; $t_1 = \text{starting negotiation process}$; $t_2 = \text{negotiation plan(s)}$; $t_3 = \text{"hand shake"}$; $t_4 = \text{proposal(s)}$; $t_5 = \text{revised proposal(s)}$; $t_6 = \text{agreement or quit}$; $t_7 = \text{ending negotiation process}$. However, the reality is more complicated. To illustrate it in details, let us introduce the PN-based model of the agent in general as well as that of more agents working in frame of MAS.

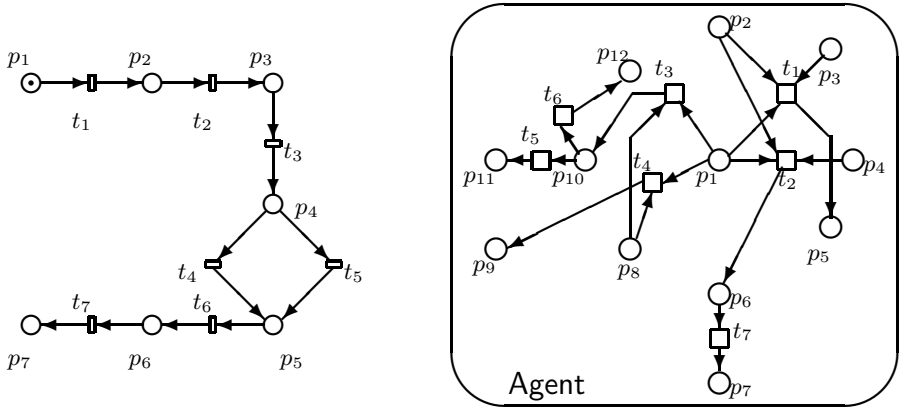


Fig. 1. The PN-based model of the coordination plan of the negotiation process in general (on the left) and the PN-based model of a general agent (on the right)

2.2 The PN-Based Model of Agents in MAS

The general structure of a general agent from the collaboration/negotiation point of view is given on the right in Figure 1 representing the PN-based model of the agent. To interpret the sense of the PN places and transitions, it is necessary to introduce: p_1 = the agent (A_1) is free; p_2 = a problem has to be solved by A_1 ; p_3 = A_1 is able to solve the problem (P_{A_1}); p_4 = A_1 is not able to solve P_{A_1} ; p_5 = P_{A_1} is solved; p_6 = P_{A_1} cannot be solved by A_1 and another agent(s) should be contacted; p_7 = A_1 asks another agent(s) to help him to solve P_{A_1} ; p_8 = A_1 is asked by another agent(s) to solve a problem P_B ; p_9 = A_1 refuses the help; p_{10} = A_1 accepts the request of another agent(s) for help; p_{11} = A_1 is not able to solve P_B ; p_{12} = A_1 is able to solve P_B . In such a case parameters of the PN-based model are the following

$$\mathbf{F}^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Analyse e.g. the situation when $\mathbf{x}_0 = (1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)^T$, i.e. the situation when the agent A_1 is free and it is asked by the agent A_2 to solve the problem $P_B = P_{A_2}$. Using the algorithm introduced in [4] we have the following quasi-adjacency matrix \mathbf{A} (its elements are the indices of the PN transitions) of the PN reachability tree and the matrix \mathbf{X}_{reach} with columns being the feasible states (the initial state and all states reachable from this initial state)

$$\mathbf{A} = \begin{pmatrix} 0 & 3 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{X}_{reach}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The reachability graph of such a PN-based model is given in Fig. 2. Such a

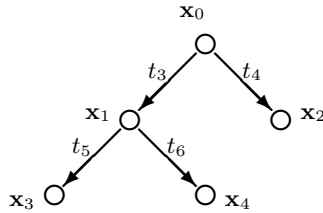


Fig. 2. The reachability graph of the agent situation

PN-based model of the agent is universal and it can be used for modelling other agents of MAS too. Namely, the same interpretation of places (however with shifted numbering p_{i+12} , $i = 1, \dots, 12$) can be used e.g for the agent A_2 . Having two agents A_1, A_2 the collaboration/negotiation of them is given in Figure 3. In case of more agents - e.g. N_A - the places numbering of the agent $k = 1, \dots, N_A$ is $p_{i+12,j}$, $i = 1, \dots, 12$, $j = k - 1 = 0, \dots, N_A - 1$. In case of several agents both the PN model and the RG will be more complex. While the model size depends on modules and their interface, the RG size depends on \mathbf{x}_0 and on the structure of model blocks.

As we can see we have $n = 24$ places and $m = 20$ transitions in the PN model of the two agents cooperation. However, the number of transitions is higher than a simple sum being $m = 14$. Namely, some transitions have to be added as an interface in order to connect both of the agents. Consequently, for two agents A_1, A_2 the matrices of the MAS parameters will have the form as follows, where the structure of the actual contact interface between the agents is given by the $(n \times 6)$ -dimensional matrix \mathbf{F}_c and $(6 \times n)$ -dimensional matrix \mathbf{G}_c

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & \mathbf{0} & \mathbf{F}_{c1} \\ \mathbf{0} & \mathbf{F}_2 & \mathbf{F}_{c2} \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 \\ \mathbf{G}_{c1} & \mathbf{G}_{c2} \end{pmatrix}$$

$$\mathbf{F}_c^T = \left(\begin{array}{cccccccccccc|cccccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right) = (\mathbf{F}_{c1}^T \mid \mathbf{F}_{c2}^T)$$

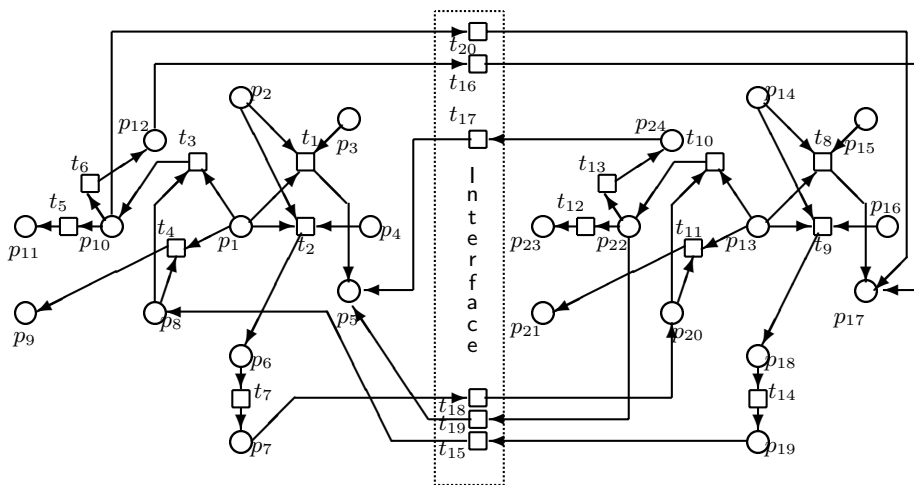


Fig. 3. The Petri net-based model of the two agents cooperation and/or negotiation by means of the interface

$$\mathbf{G}_c = \left(\begin{array}{cccccccccccc|cccccccccccc}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 \\
 \end{array} \right) = (\mathbf{G}_{c_1} \mid \mathbf{G}_{c_2})$$

In general, for agents $i = 1, 2, \dots, N_A$ the structure of matrices is the following

$$\mathbf{F} = \left(\begin{array}{cccc|c}
 \mathbf{F}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{F}_{c_1} \\
 \mathbf{0} & \mathbf{F}_2 & \dots & \mathbf{0} & \mathbf{F}_{c_2} \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{F}_{N_A-1} & \mathbf{0} & \mathbf{F}_{c_{N_A-1}} \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{F}_{N_A} & \mathbf{F}_{c_{N_A}}
 \end{array} \right); \quad \mathbf{G} = \left(\begin{array}{cccc|c}
 \mathbf{G}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\
 \mathbf{0} & \mathbf{G}_2 & \dots & \mathbf{0} & \mathbf{0} \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{G}_{N_A-1} & \mathbf{0} \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{G}_{N_A} \\
 \hline
 \mathbf{G}_{c_1} & \mathbf{G}_{c_2} & \dots & \mathbf{G}_{c_{N_A-1}} & \mathbf{G}_{c_{N_A}}
 \end{array} \right)$$

$$\mathbf{B} = \left(\begin{array}{cccc|c}
 \mathbf{B}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_{c_1} \\
 \mathbf{0} & \mathbf{B}_2 & \dots & \mathbf{0} & \mathbf{B}_{c_2} \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_{N_A-1} & \mathbf{0} & \mathbf{B}_{c_{N_A-1}} \\
 \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_{N_A} & \mathbf{B}_{c_{N_A}}
 \end{array} \right) = (\text{blockdiag}(\mathbf{B}_i)_{i=1, N_A} \mid \mathbf{B}_c)$$

$$\mathbf{B}_i = \mathbf{G}_i^T - \mathbf{F}_i; \quad \mathbf{B}_{c_i} = \mathbf{G}_{c_i}^T - \mathbf{F}_{c_i}; \quad i = 1, 2, \dots, N_A; \quad \mathbf{F}_c = (\mathbf{F}_{c_1}^T, \mathbf{F}_{c_2}^T, \dots, \mathbf{F}_{c_{N_A}}^T)^T$$

$$\mathbf{G}_c = (\mathbf{G}_{c_1}, \mathbf{G}_{c_2}, \dots, \mathbf{G}_{c_{N_A}}); \quad \mathbf{B}_c = (\mathbf{B}_{c_1}^T, \mathbf{B}_{c_2}^T, \dots, \mathbf{B}_{c_{N_A}}^T)^T$$

with $\mathbf{F}_i, \mathbf{G}_i, \mathbf{B}_i, i = 1, 2, \dots, N_A$, representing the parameters of the PN-based model of the agent A_i , and with $\mathbf{F}_c, \mathbf{G}_c, \mathbf{B}_c$ representing the structure of the actual contact interface between the agents cooperating in MAS. Using the own graphical tool (developed during last years by master students) for drawing the model by means of icons the properties of the model can be tested. The reachability tree (RT) being the most important instrument on this way can be developed in the graphical form. Starting from the initial state $\mathbf{x}_0 = (\sigma_{p_1}, \dots, \sigma_{p_n})^T = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0)^T$, i.e. from the state where only six places have nonzero marking, namely $\sigma_{p_1} = 1, \sigma_{p_2} = 1, \sigma_{p_3} = 1, \sigma_{p_{13}} = 1, \sigma_{p_{14}} = 1, \sigma_{p_{16}} = 1$, we can draw the reachability graph of the PN-based model of two agents cooperation given in Figure 4. It has $N = 13$ nodes representing the feasible states $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{12}\}$ of the system state space.

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 6 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The edges express feasible paths (trajectories) among the corresponding states. Such an approach to analysing DEES is very comfortable and user friendly. However, from the ergonomy point of view, the observed area as well as the number of variables which the human operator is able to recognize are limited. Hence, more formal approaches are found for DEES analysis and control. Thus, large-scale mathematical models can be handled too (e.g. by means of Matlab).

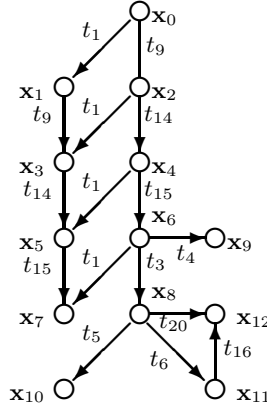


Fig. 4. The reachability graph of the two agents cooperation

3 Analyzing the Model and the Control Synthesis

In [8] the Matlab procedure enumerating the $(n_{RT} \times n_{RT})$ -dimensional *quasi-functional* adjacency matrix \mathbf{A} of the RG and the space of the reachable states in the form of the $(n \times n_{RT})$ -dimensional matrix \mathbf{X}_{reach} was presented. The matrices fully characterize the RG. The \mathbf{X}_{reach} columns (the RG nodes) are the PN state vectors $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_{RT}-1}$ reachable from the initial state \mathbf{x}_0 . The inputs of the procedure are the matrices \mathbf{F}, \mathbf{G}^T and the initial state vector \mathbf{x}_0 . The DEDS control synthesis based on the simple idea was also presented in [8]: (i) developing the straight-lined reachability tree (SLRT) from the given initial state \mathbf{x}_0 towards the prescribed terminal state \mathbf{x}_t ; (ii) developing the backtracking reachability tree (BTRT) from \mathbf{x}_t towards \mathbf{x}_0 however, directed to \mathbf{x}_t ; (iii) intersecting both the SLRT and BTRT. All of the steps are performed numerically in Matlab. To compute SLRT the graph-based model $\mathbf{X}(k+1) = \mathbf{A}_k^T \cdot \mathbf{X}(k), k = 0, 1, \dots, N-1$ is used while for computing BTRT the model $\mathbf{X}(k-1) = \mathbf{A}_{k-1} \cdot \mathbf{X}(k), k = N, \dots, 1$, with $N \leq n_{RT} - 1$, is used. \mathbf{A}_k is the functional (k -variant) adjacency matrix, $\mathbf{X}(k), k = 0, 1, \dots, n_{RT} - 1$ are the hyperstate vectors - the unit vectors with only one nonzero element equal to 1 on the position corresponding to the index of the \mathbf{X}_{reach} column. The hyperstate vector $\mathbf{X}(0)$ corresponding to \mathbf{x}_0 has 1 in its first position while the hyperstate vector corresponding to the terminal state \mathbf{x}_t appearing e.g. in the column j of \mathbf{X}_{reach} will have 1 in the position j - i.e. it will be the hyperstate vector $\mathbf{X}(j)$. After intersection of both SLRT and BTRT the feasible trajectory (or several ones) starting from \mathbf{x}_0 and finishing in \mathbf{x}_t is (are) obtained. Having the set of feasible trajectories, the most suitable one (satisfying imposed control task specifications) can be chosen. More details about the procedure can be found in [5], [8]. The graphical tool GraSim was developed for DEDS modelling, analyzing and the control synthesis. The input of the tool is the RG of the PN model. It is created by means of clicking (by the mouse) on icons representing the RG nodes and edges as well as the marks for designating the initial and terminal states. On its output the tool yields (in the

graphical form) the system trajectories from a given initial state to a prescribed terminal one. The trajectories can be analyzed one after another.

4 Conclusions

The alternative PN-based approach to modelling, analysis and control of the agent interactions in MAS was presented in this paper. It yields the results in analytical terms as well as graphically. The PN-based model of the interaction proces is created in analytical terms, the RG and the space of reachable states are generated. Feasible state trajectories are found by means of the mutual intersection of both the SLRT and BTRT. The graphical tool GraSim was developed to automate the control synthesis.

References

1. Brams, S.J., Kilgour, D.M.: Fallback bargaining. *Group Decision and Negotiation* **10** (2001) 287–316
2. Čapkovič, F.: Modelling and Control of Discrete Event Dynamic Systems. BRICS Report Series, RS-00-26, University of Aarhus, Denmark (2000) 58 p.
3. Čapkovič, F.: Control synthesis of a class of DEDS. *Kybernetes. The Int. Journ. of Sys. and Cybern.* **31** No 9/10 (2002) 1274–1281
4. Čapkovič, F.: The generalised method for solving problems of DEDS control synthesis. *Lecture Notes in Artificial Intelligence* **2718** (2003) 702–711
5. Čapkovič, F.: An application of the DEDS control synthesis method. *Lecture Notes in Computer Science* **3038** Part III (2004a) 528–536
6. Čapkovič, F.: DEDS control synthesis problem solving. In: *Proc. 2nd IEEE Conf. on Intelligent Systems, Varna, Bulgaria*. IEEE Press (2004b) 299–304
7. Čapkovič, F., Čapkovič, P.: Petri net-based automated control synthesis for a class of DEDS. In: *Proc. 2003 IEEE Conf. on Emerging Technologies and Factory Automation*. Lisbon, Portugal. IEEE Press (2003) 297–304
8. Čapkovič, F.: An application of the DEDS control synthesis method. *Journal of Universal Computer Science* **11** No 2 (2005) 303–326
9. Hung, P.C.K., Mao, J.Y.: Modeling e-negotiation activities with Petri nets. In: *Proc. 35th Hawaii Int. Conf. on System Sciences*. Big Island, Hawaii, Vol. 1. IEEE Computer Society Press (CD-ROM) (2002) 10 p.
10. Looney, C.G.: Fuzzy Petri nets for rule-based decision-making. *IEEE Trans. Syst. Man Cybern.* **18** No 1 (1988) 178–183
11. Nowostawski, M., Purvis, M., Cranefield, S.: A layered approach for modelling agent conversations. In: *Proc. 2nd Int. Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, Montreal, Canada (2001) 163–170.
12. Purvis, M., Cranefield, S., Nowostawski, M., Ward, R., Carter, D., Oliveira, M.A.: Agentcities interaction using the opal platform. In: *Proc. Workshop on Agentcities: Research in Large-Scale Open Agents Environments*, 1st Int. Joint Conf. on Autonomous Agents and Multi-Agent. Bologna, Italy. CD-ROM. (2002) 5 p.
13. Thompson, L.: *The Mind and Heart of the Negotiator*. Prentice-Hall Inc. (1998)
14. Yen, J., Yin, J., Ioerger, T.R., Miller, M.S., Xu, E., Volz, R.A.: Cast: Collaborative agents for simulating teamwork. In: *Proc. 17th Int. Joint Conf. on Artificial Intelligence - IJCAI'2001*, Seattle, USA (2001) 1135–1142.

A Semantic-Driven Cache Management Approach for Mobile Applications

Guiyi Wei, Jun Yu, Hanxiao Shi, and Yun Ling

Zhejiang Gongshang University, Hangzhou, 310035, P.R. China
weiguiyi@tom.com, {yj, hxshi, yling}@mail.zjgsu.edu.cn

Abstract. With the development of wireless communication technology, mobile business become more and more popular. Using GPRS or WAP protocols, the wireless devices can connect to the Web servers, retrieve information from the online databases and run special application programs. Because of the limitation of the wireless communication and mobile computing environment, it is difficult to improve the execution efficiency for the program that located in mobile devices. To solve the problem, introducing the cache mechanism is the major and effective method. But the traditional cache model can not achieve an acceptable cache hit ratio. The semantic caching is particularly attractive in a mobile business environment, due to its content-based reasoning ability and semantic locality. In semantic-driven cache model, only the required data is transmitted to wireless device. In this paper we propose an application-oriented semantic cache model. It establishes an semantic associated rule-base according to the knowledge of application domains, makes use of the semantic locality for data prefetching, and adopts a Two-level LRU algorithm for cache replacement. Several experiments demonstrate that the semantic-driven cache model can achieve higher hit ratio than traditional models.

1 Introduction

E-business is being used to overcome the geographic limitation and improve the efficiency of commercial activities[1]. It plays a more and more important role in world commerce for its high quality of services. At the same time, the popularization of the wireless devices, such as cell-phone, PAD, laptop and etc., has enabled the mobile business becoming realized. Some mobile applications have come into our lives. People can play network games with connecting to an internet host through wireless network. Customers can perform online payment through remote banking system. Individual investors can browser the instant quotations of the stock market and make stockbroking. The travelers can book hotel with their wireless communication devices. Therefore, the development of mobile business brings the revolutionary changes to our future lives.

Using GPRS or WAP protocols, the wireless devices can connect to the Web servers, retrieve information from the online databases and run special application programs. In some case, the mobile device need to run some special programs with

amount of data, such as reporting, analysis and etc., rather than only run a Web browser. Because of the limitation of the wireless communication and mobile computing environment, it is difficult to improve the execution efficiency for the program that located in mobile devices. To solve the problem, introducing the cache mechanism is the major and effective method. But the traditional cache model can not achieve an acceptable cache hit ratio. The major method to solve this problem is to adopt the cache management with prefetched data mechanism[4,5]. The related research attempted to apply command-driven and data-driven methods for data prefetching. These methods partly alleviate the problem of the low bandwidth and high delays in the mobile computing environment[6,7]. Lacking the semantic understanding for application domains, the production of associated rules, the method of data prefetching and cache replacement are blindness and maybe invalid. It would waste limited bandwidth. And the hit-ratio and validity of cache are not observably improved. The semantic caching is particularly attractive in a mobile business environment, due to its content-based reasoning ability and semantic locality. In semantic-driven cache model, only the required data is transmitted to wireless devices. In this paper we propose an application-oriented semantic cache model. It establishes an semantic rule-base according to the knowledge of application domains, makes use of the semantic locality for data prefetching, and adopts a Two-level LRU algorithm for cache replacement.

2 Mobile Processes Model

In the mobile process model (depicted in Fig.1), the mobile application consists of a client component and a server component. The client program resides in mobile terminals, and the server program and database are reside in the online web hosts. They are connected with wireless communication network [2,3].

Before the client program execute, it is divided into a key process and several relative independent sub processes or threads, which are responsible for query generating and data processing. A supervisor process of local operating system take charge of coordination of these sub processes and the key process. It schedules processes with consideration of the processor's workload, state of the network and data ready state in the cache.

The cache of the mobile terminal is divided into a basic cache and a prefetched data set. The basic cache accommodates the data produced with the temporal locality principle. The prefetched data set stores the data prefetched with the semantic locality principle, which will be used when the basic cache mismatch (not hit). The prefetched data set is replaced using associated rules which derived from the knowledge of special application domain.

The server side provides source data management services. The data service is divided into several lightweight transactions. Each lightweight transactions is response to one sub-process of mobile client. This will alleviate the impact of the conceivable network linkage failure.

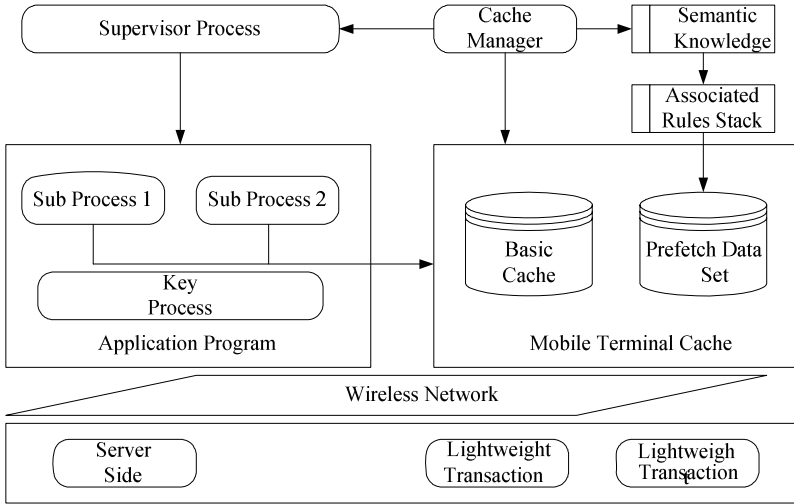


Fig. 1. The process model in the mobile application environment

3 Cache Management Language (CML)

3.1 Definition of CML

The CML language should express three aspects of mobile cache management, the data correlativity, the data dependency and the critical point of replacement. It is similar to the profile language in [10]. The CML instruction is executed by a grammar interpreter without compiling. In CML, there are three basic elements, *domain*, *utility* and *operation*. The *domain* defines a data set, the *utility* defines relations of data in *domains*, and *operation* is action performed when certain conditions are satisfied.

3.2 Prefetch Algorithms

At present, the mobile network communication is characterized by low bandwidth and high latency. It is a challenge to mobile applications in data communication. The mobile terminal usually uses the method of demanded data prefetching to handle these problems. The prefetch algorithm for mobile cache can be classified two types, preemptive and non-preemptive. Our cache model mainly aimed the mobile business application that is interactive program and performs complex data processing workflow, so the preemptive prefetch algorithm is adopted.

The preconditions of prefetch algorithm include: O is a finite set of data object; o is any element of O ; $s(o)$ is the space of data object o ; C is the capacity of semantic cache. pr is the possibility of preemption; $S = \langle o_1, o_2, \dots, o_n \rangle$ is a subset of O that satisfy the conditions of formulas (1)-(5); p is an operation instance described with CML language, g is a utility function; fp is weights calculation function.

$$\sum_{i=1}^n s(o_i) \leq C \quad (1)$$

$$\text{Max } g_{p, pr}(O) \quad (2)$$

$$g_{p, pr}(S) = e_1 + e_2 \quad (3)$$

$$e_1 = (1 - pr)(f_p(S)) \quad (4)$$

$$e_2 = pr \left(\frac{(\sum_{i=2}^n s(o_i) \cdot f_p(\{o_1, \dots, o_{n-1}\}))}{s(o_1) + \dots + s(o_n)} \right) \quad (5)$$

The prefetch algorithm is an ameliorated greedy algorithm according to formulas (1)-(5). It improved the the performance of common greedy algorithm in cache management.

3.3 Semantic Cache Model

The semantic cache usually consists of two parts, the index part and the content part [8, 9]. The content part consists of pages that store the semantic segments, each page has a unique page number for identification. A page number can be mapped to a physical page in the cache. The semantic segment can be represented as formula (6): S_R is the source data objects participated prediction and the relationships among them; S_A is the common attributes of a data object set; S_p is the data objects prediction using associated rules; S_C is the data content of semantic segment; S_{PG} is the page that contains the semantic segment; S_T is the latest accessed time of the semantic segment.

$$S = \langle S_R, S_A, S_p, S_C, S_{PG}, S_T \rangle \quad (6)$$

The emphasis of semantic cache organization is its replacement policy. We use an improved two-level LRU algorithm for semantic cache replacement (detailed described in section 4 of this paper). The operation of semantic cache is depicted as Fig.2.

3.4 Associated Rules Generation

We define semantic knowledge set $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}$, data item $F = \{i_1, i_2, i_m\}$. S_i is a set of semantic data item F which adapted to semantic knowledge p_i . If X and Y are two data item sets, $X \subseteq S_i$, $X \subset F$, $Y \subset F$, and $X \cap Y = \Phi$, then $X \Rightarrow Y$ is an associated rule.

$Sup(X)$ is the supported degree of $X \subseteq S_i$, $Sup(R) = Sup(\{X, Y\})$ is the supported degree of formula $R: X \Rightarrow Y$. $Conf(R) = \frac{Sup(\{X, Y\})}{Sup(X)} * 100\%$ is the confidence degree.

Before an associated rule generating, a minimum supported degree $min-sup$ and a minimum confidence degree $min-conf$ are set. The generation algorithm automatically generate an X with $Sup(X)$ is not smaller than $min-sup$. Then it produce a Y according to X , and compute R with $Conf(R)$ not smaller than $min-conf$.

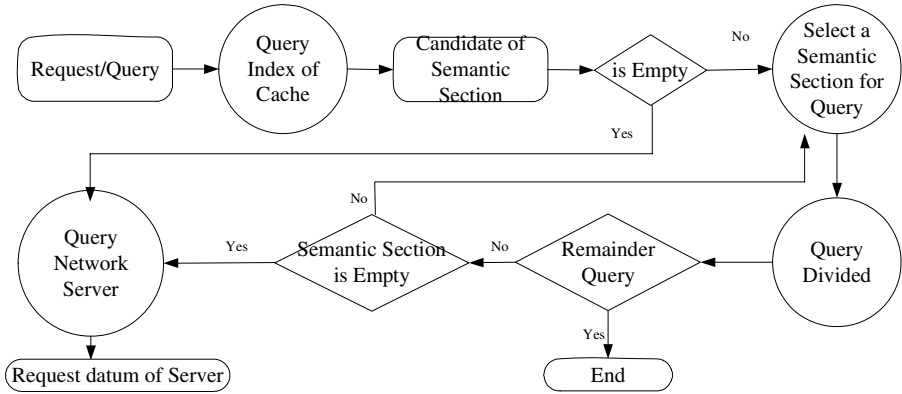


Fig. 2. Operation of semantic Cache

4 Replacement Algorithm

Traditional cache replacement policies are based on temporal locality or spatial locality. In order to preserve data items that are most likely to be accessed in the future and improve cache hit-ratio, our cache model adopted semantic locality for cache replacement. It use a two-level LRU(least recently used) replacement policy to predict the future access probability by analyzing the semantic locality with associated rules.

The first level LRU generates candidate page set using temporal locality policy while cache mismatch. The second level LRU uses semantic locality policy to choose page from candidate page set. The pages selected by second level LRU will be swapped out.

Algorithm. Two-Level LRU Cache Replacement

```

while ( page fault occur)
{
  cs <- generate candidate page set using temporal
        locality based LRU algorithm (First Level LRU);
  while ( cs is not NULL)
  {
    sd <- max value;
    sp <- NULL; selected page;
    p <- fetch next page from cs;
    psd <- compute the support degree with p and semantic
           associated rules;
    if psd < sd then
      {sd <- psd; sp <- p;}
    move p from cs;
  }
  discard p from semantic cache;
  add free page;
}

```

5 Experiments

A web based mini ERP system and a mobile phone are used to simulate a mobile business application environment. The ERP system includes a sale management module and a data analysis module. These modules are interactive programs, and run with volumes of data. The mobile phone can adapt to embedded memory with variable volume sizes. The mobile device communicates with online application server using GPRS protocol. Before the client of the application executes, its kernel code and necessary data have been reside in the mobile terminal. The kernel code will run as kernel process as depicted in the figure 1. It surely can be downloaded and updated online. During the execution of the program, additional function codes of the application may be downloaded as demanded, and volumes of data should be transferred from and to between client and server.

Based on the implemented prototype of our proposed model and its algorithms, some performance evaluations are carried out.

In the first experiment, we split whole client memory up into two parts equally (50% basic cache and 50% prefetched data set). The replacement of basic cache uses the first level LRU algorithm with temporal locality. The replacement of prefetched data set uses the second level LRU algorithm with semantic locality. The changes of cache hit ratio according to different cache size are illustrated in Fig.3.

In the second experiment, whole cache size is fixed to 1 megabytes, prefetch data set size is changed dynamically from 40% to 90% . Additionally, with the same cache size, a data-driven method and a command-driven method are evaluated to compare with the semantic-driven method. Tests of data-driven method and command-driven method use same replacement algorithm for two parts of caches. The changes of cache hit ratio of different methods are illustrated in Fig.4.

As depicted in figure 4, we can see: (1) the semantic-driven method can achieve higher hit ratio with more prefetch set proportion given initially, the highest hit ratio appears with the proportion of 75% approximately, and then hit ratio declines gradually; (2) the semantic-driven method can achieve higher hit ratio than command-driven method and data-driven method. Further experiments will be done to refine semantic-driven model and its algorithms.

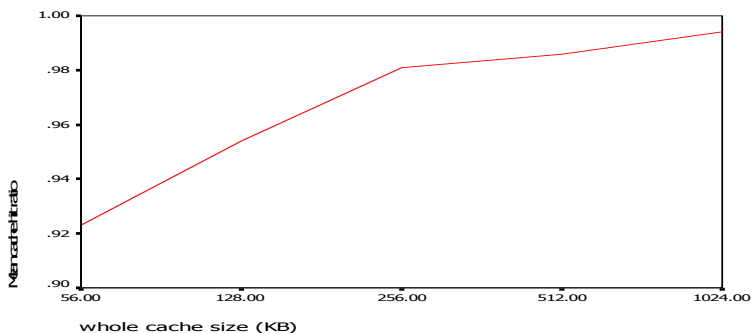


Fig. 3. Cache hit ratio according to different cache size with fixed division

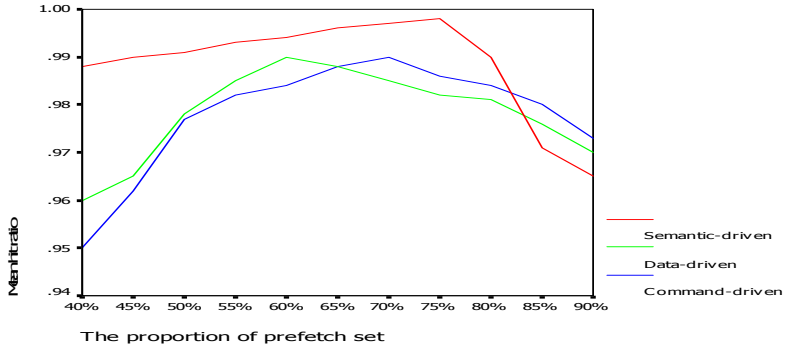


Fig. 4. Cache hit ratio according to different methods with dynamic division

6 Conclusions

In this paper, we proposed an mobile process model suitable for mobile client program, and discussed a semantic-driven model for cache management of mobile devices in mobile business environment. The design and implementation of a prefetch algorithm and a two-level LRU cache replacement algorithm are detailedly illuminated. The results of corresponding experiments demonstrated that the semantic-driven model (1) is effective and efficient, (2) can achieve higher hit ratio than data-driven and command-driven methods.

Acknowledgments

The authors wish to thank the Natural Science Foundation of Zhejiang Province, P. R. China for the National Science Fund under grant Number Y105356. We would like to thank CRMB (the Center for Research in Modern Business, Zhejiang Gongshang University) for they partially supported this research. We would also like to thank our referees for their helpful comments and suggestions.

References

1. André Köhler, Volker Gruhn: Analysis of Mobile Business Processes for the Design of Mobile Information Systems. Proceedings of E-Commerce and Web Technologies, EC-Web 2004, LNCS3182, Springer, (2004)
2. Noor, N. M. M., Papamichail, K. N., Warboys, B.: Process Modeling for Online Communications in Tendering Processes. Proceedings of the 29th EUROMICRO Conference. IEEE Computer Society, (2003)17-24
3. Ritz, T., Stender, M.: Modeling of B2B Mobile Commerce Processes. 17th International Conference on Production Research ICPR-17. Virginia Tech, Blacksburg, (2003)
4. D. Barbara and T. Imielinski: Sleepers and Workaholics: Caching Strategies for Mobile Environments. ACM SIGMOD, (1994)1-12

5. G. Cao: A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. *IEEE Transactions on Knowledge and Data Engineering*. ACM MobiCom'00, (2000)
6. H. Song and G. Cao: Cache-Miss-Initiated Prefetch in Mobile Environments. *IEEE International Conference on Mobile Data Management (MDM)*, (2004)
7. V. Grassi. Prefetching Policies for Energy Saving and Latency Reduction in a Wireless Broadcast Data Delivery System. In *ACM MSWIM 2000*, Boston MA, (2000)
8. R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules. *Proc. 20th Int. Conf. Very Large Data Bases*. Morgan Kaufmann (1994)487–499
9. R. Agrawal, Tomasz Imielinski, and Arun Swami: Mining Association Rules Between Sets of Items in Large Databases. In *Proc. of the ACM SIGMOD Conference on Management of Data*. (1993) 207–216
10. Mitch Cherniack, Eduardo F. Galvez, Michael J. Franklin, Stan Zdonik: Profile-Driven Cache Management. *Proceedings of International Conference on Data Engineering (ICDE)*, Bangalore, India, (2003)

Fault Tolerance Mechanism of Agent-Based Distributed Event System

Ozgur Koray Sahingoz¹ and A. Coskun Sonmez²

¹ Air Force Academy, Computer Engineering Department, Yesilyurt, Istanbul, Turkey
sahingoz@hho.edu.tr

² Yildiz Technical University, Computer Engineering Department, Yildiz, 34349,
Istanbul, Turkey
acsonmez@ce.yildiz.edu.tr

Abstract. Event based system development is increasingly becoming popular for large-scale and heterogeneous distributed platforms because it helps diminishing software dependencies, and enhancing system integration and evolution. The architecture of an event based system should be tolerant to error and network fallout especially in dispatching service. Throughout the entire design of event based systems, fault-tolerance mechanism plays very important role in developing large scale middleware. This is a crucial quality of service where node failures are frequent in wide area networks with many brokers. In this paper, we address fault tolerance mechanism of the agent based distributed event system where events are responsible for determining their own paths, in the case of link and broker failures. This mechanism is achieved by dynamically configuring new paths at run time for making the system more scalable and robust on a global scale.

1 Introduction

Event based systems consist of distributed components which communicate through the exchange of event messages which are defined as simple messages, such as records, tuples, or simple objects. Middleware using an event based communication model is appropriate to address the requirements of distributed applications for large scale and heterogeneous environments which require a less tightly coupled communication relationship between their components.

Event based communication generally implements the publish/subscribe model which is very well suited for connecting loosely coupled large-scale applications in the Internet [1-3]. In this model, receivers of messages (subscribers) express their interest by subscribing to a class of events, and they are asynchronously notified if a sender (publisher) publishes an event which matches the subscription. In this way the model allows a flexible n-to-m communication among the communicating parties. The core component of the middleware, the dispatching system, is responsible for collecting subscriptions and advertisements. After that the dispatching system forwards event messages from publishers to subscribers and in doing so achieves a high degree of decoupling among the communicating parties.

Faults and failures are inevitable in a distributed system built over a wide-area network and they should be tolerated for continuing system transactions. Fault-tolerance mechanisms in distributed event systems can cope with different kinds of failures in system middleware and especially integrated with the routing algorithm. This type of fault tolerance results a scalable and robust system.

In this work, we are concerned with the fault tolerance of node or link error in the Agent Based Distributed Event System-(ABDES) [10] in which events are represented by a mobile agent, called as *agvent*, which is treated as a first class citizen of the system. Autonomy and mobility features are given to *agvents* to select and travel between system components. We assume that brokers (called as *agvent servers* in our system) in this dispatching service of ABDES can fail by crashing and that the failure of a broker is eventually detected by all its neighbor brokers by using well-know failure detection techniques, like heartbeats or by detecting when there is a necessity. A failed broker can cause a gap in the event dissemination tree. To heal the tree, the broker which detects the failure re-routes the subscriptions, advertisements and events (called as *agvents*) via an alternative route.

The rest of this paper is organized as follows. In the next section, we present fault tolerance mechanisms of event based systems. Next, dispatching mechanism of ABDES is explained shortly in Section 3. Fault tolerance mechanism of ABDES is presented in Section 4 and finally, we present our conclusion and plans for future work in Section 5.

2 Related Works

Researches on fault tolerance mechanism of event based systems are concentrating on two main topics; *Self Stabilization* and *Reconfiguration*. In [4, 5] authors assume that some link may disappear and others appear elsewhere, because of changes in the underlying. *Reconfiguration* in this case means fixing routing tables entries no longer valid after the topology change. Trigger for such reconfiguration is on disappearance of one or more links between brokers, and possibly the appearance of new ones. *Self-stabilization* is an optimistic way of looking at system fault tolerance and scalable coordination, because it provides a built-in safeguard against transient failures that might corrupt the data in a distributed system. Some event systems [6, 7] use self-stabilization in the broker network by discarding broken and outdated information about neighbors. This methodology is used for synchronizing routing tables because of message loses and is accomplished by the use of leases. Both these models have a side effect as significant increase of the network traffic. Therefore some researches [8] have been done to minimize traffic overhead of the system.

There are only a few researches [9] in the literature regarding node and communication link failures. In these researches, when a broker fails, the another broker can take over the role and publish/server model can continue its transactions without any interruption and when the communication link is disconnected data transfer is delayed until the link is reconnected. But, if the link is not reconnected then messages over this link cannot be dispatched.

3 Dispatching Mechanism of ABDES

The historical development of publish/subscribe systems has followed a line which has evolved from channel-based systems, to subject-based systems, next content-based systems and finally object/type based systems. We thought the next step should contain agents in the system and developed an *Agent Based Distributed Event System-ABDES* [10] which defines events (called as *agvents-AGent eVENTs*) as first class members of the system.

ABDES combines the advantages of publish/subscribe communication and mobile agents into a flexible and extensible distributed execution environment. The major novelty of the model is that an event is represented as a mobile agent and given autonomy and mobility features to select and travel between system components. The benefits of the model are; reduced network load, higher adaptability by allowing dynamic changes in system configuration, information hiding, asynchronous communication and flexibility of agent based execution. We think the new model will serve as an effective choice for several information oriented applications, such as e-commerce, information retrieval, publication dispatch systems, distributed software and virus definitions updating.

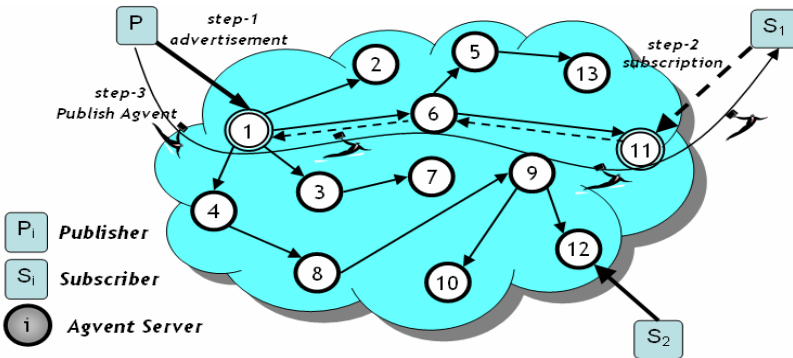


Fig. 1. Dispatching Mechanism of Agent Based Distributed Event System

Dispatching mechanism of ABDES is detailed in [11] and shortly depicted in Fig. 1. In ABDES, Firstly, a publisher advertises its agent types which he will publish to the system (step-1). This advertisement is dispatched to all agent servers in the dispatching service via a broadcast message. If a subscriber decides to subscribe on an agent type, it sends a subscription message to its connected agent server, and this message is dispatched to all relevant agent servers by the system (step-2). Whenever a publisher observes an event, it creates and sends an agent to the agent server which it is connected (step-3). When the agent arrives on the agent server, it starts to execute its pre-specified code to select its targets (neighbor agent servers and/or registered subscribers) according to the information present in the subscription table.

Faults in ABDES can be in each of these three dispatching operations (advertisement dispatching, subscription dispatching and agent dispatching).

Therefore we detailed the fault tolerance mechanism of the system on these dispatching mechanisms.

4 Fault Tolerance Mechanism

Dispatching service in most of the event systems is implemented in a distributed fashion to increase the scalability of the system. Dispatching service is constituted as a set of brokers inter-connected in an overlay network. Brokers cooperatively route the messages and events issued by the clients connected to them. Although the topology of brokers based on a graph topology is starting to appear (as we implement), most available event systems are based on a tree topology, because this simplifies routing and provides a high degree of scalability.

The architecture of event based systems should be tolerant to error and network fallout especially in dispatching service. Therefore ABDES has a fault tolerance mechanism that can cope with different kinds of failures in the middleware and it is integrated with the routing algorithm resulting in a scalable and robust system.

In this study we investigate fault tolerance mechanism in the case of broker and communication link failures. When a link is removed, each of its endpoints is no longer able to route messages and events to the other partition and when a broker is down messages cannot be sent over it. Therefore we developed a fault tolerance mechanism by rerouting of the messages over healthy nodes and links to their clients. Fault tolerance mechanism of ABDES is thought in three dimensions for faults in *advertisement, subscription and agent dispatching*.

4.1 Fault Tolerance in Advertisement Dispatching

In ABDES, advertisements are used to make agent types visible to all the participants of the system. Advertisement forwarding, limits the overhead of subscription dispatching by spreading knowledge about agents throughout the system. When an agent server receives an advertisement from one of its neighbors, not only it stores the associated agents behaviors and properties into its advertisement table, but it also forwards it to all the remaining neighbor agent servers, thereby forming a tree to reach all agent servers. This process effectively sets up routes for subscriptions through *the reverse path* followed by advertisements.

The technique of flooding is the simplest approach to implement the advertisement propagation in dispatching service. Here, every agent server forwards an advertisement that is produced by one of its local clients to all of its neighbors, and if an agent server receives an advertisement from a neighbor, it simply forwards it to all other neighbor agent servers.

In dispatching of an advertisement, if there is a node or link error and one of agent servers (or more) is disconnected, then this agent server cannot get advertisement messages. This fault causes a disadvantage on routing this advertisement message not to all relevant agent servers. When this agent server is online or link is repaired, the agent server synchronizes its advertisement table by controlling the neighbor agent servers' knowledge bases. If there are some unsynchronized advertisement messages, these are received from them and forwarded to other neighbor agent servers (if there

is any). This mechanism is also used for adding new agent as a new neighbor to one already in the system. As is to be expected, improving scalability can be seen as more agent servers are added to the system. Scalability of brokers in an event system is important today. As the Internet continues to grow in popularity and size, dispatching service without good scalability will result in performance drop or live-lock. When a new agent server joins to the system, agent servers gets the advertisement table of the one of its neighbor and uses it.

4.2 Fault Tolerance in Subscription Dispatching

A subscription expresses the subscriber's interest in the occurrence of specific events. It is a logical expression that provides the ability to select a subset of events based on their content or type. The compatibility between subscriptions and advertisements is also important because, in setting up the routing information, the event service takes advertisements into account to see if they are relevant to any subscription. As usage of advertisements to set up routes for subscriptions, subscription messages are used for setting routes for agents. During the propagation of subscription messages, each agent server behaves as a subscriber with respect to relevant neighbors. Consequently, each of them records the filter associated with the subscription in its own subscription table and forwards it towards the root agent servers which send advertisement messages. This process effectively sets up routes for agents through the reverse path followed by subscriptions.

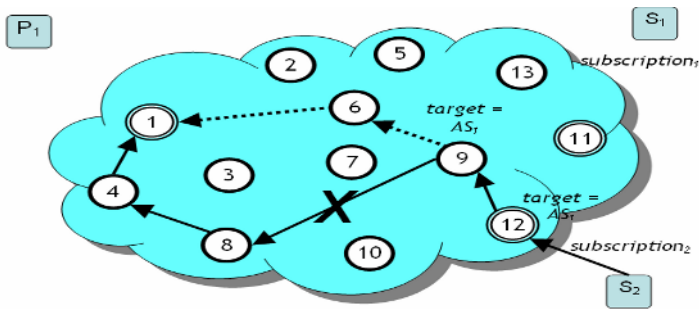


Fig. 2. Fault Tolerance in Subscription Dispatching

In dispatching of subscription, if there is a node or link error in the system (as shown in Fig. 2) then agent server which detected node/link error should chose an alternative route and forward message on this route. For example if there is a link error between AS9 and AS8 then AS9 cannot forward subscription message over AS8. In this case AS9 can detect the link failure and checks its *Alternative Routing Table (ART)* for an alternative route to AS1. ART is maintained by each agent server and composed by every incoming message (advertisements, subscription and agents) which also contains their routes to reach their clients. In this failure case the solution can be in two forms;

- If AS9 contains an alternative route between AS9 and AS1, it forwards this subscription message on this route (see Fig 2). Other agent servers use these messages for updating their ARTs.
- If there is not any routing information between AS9 and AS1 then AS9 should search an alternative route for it. To achieve this it broadcast a subscription message whose target is set as AS1 to all neighbor agent servers. This message is not evaluated as a subscription message except AS1.

In case of adding new agent server to dispatching service, advertisement table of this new agent server is copied (synchronized) from one of neighbor agent servers. But there is no need to copy the subscription table of the neighbors. Subscription table is composed according to needs of the connected subscribers. In the case of adding an agent server, two components can join to the system over this agent server; *publishers* and *subscribers*.

- If a subscriber connects to system over this new agent server, it sends a subscription message to its connected agent server and this message is forwarded according to copied advertisement table. After that the subscription table is updated.
- If a publisher connects to the system over this new agent server, it sends an advertisement message to the agent server and this message is broadcasting to all agent servers in dispatching service. Agent servers which have connected subscribers interested in this type of agents reply this advertisement message and forward a subscription message to this newly added agent server.

4.3 Fault Tolerance in Agent Dispatching

An agent is a collection of code and data that migrates through ABDES. A significant capability of an agent is its ability to discover target nodes, namely subscribers which demanded to be notified of the occurrence of an event, and to route itself to these subscribers. This is accomplished by enabling the published agent to search the knowledge base of an agent server, select the registered subscribers, clone itself and send each agent clone to a subscriber on the selected list. An agent

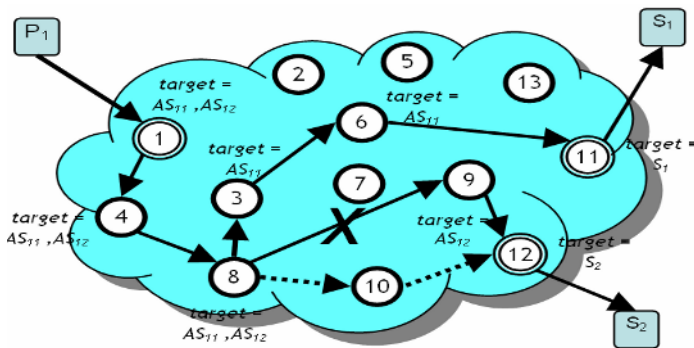


Fig. 3. Dispatching Agents

determines its own path through the network, utilizing the minimal set of facilities provided by agvent servers.

Route of agvents are decided by subscription messages (reverse path of subscription messages). As shown in Fig. 3, the published agvent should be dispatched to S1 and S2. Agvents forward themselves to these destination nodes. After agvent reaches AS8, a clone of the agvent is created and one is forwarded over AS3, and the other is forwarded over AS9. Because there is a link error between AS8 and AS9, the agvent cannot reach to AS9. Therefore AS8 search an alternative route from its ART. If it finds a route, it forwards this agvent over new route (as depicted in Fig. 3).

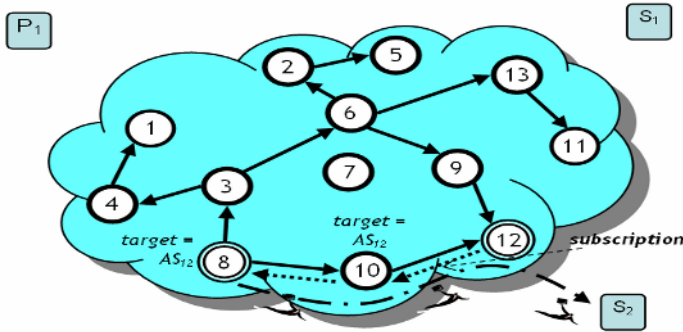


Fig. 4. Fault Tolerance in Agvent Dispatching

Main problem will remain exist if there is not any alternative route. In this case we cannot broadcast the agvent as we do in fault tolerance of subscription dispatching. An agvent is a collection of code and data. Therefore its size could be from 4-6 Kilobytes to 8-10 Megabytes (or more) according to its data block. Consequently, we have to establish a new route to sent message from breaking node to destination node. To achieve this we make a small part of previous messaging processes as follows.

- An advertisement is broadcasted to reach the target agvent server (sampled in Fig.4. as AS12). This message is taken into consideration only by the destination server (AS12). Other servers use this message only for updating their ART.
- Destination agvent servers reply this advertisement message and forward the subscription message about that type of agvent. After this subscription message reached to breaking server, an alternative route is set as reverse path of the subscription message.
- Agvent Server re-route the waiting agvent(s) to destination server(s) over this new route.

5 Conclusion

This paper presents the fault tolerance mechanism of the agent based distributed events systems, called as ABDES, which combines the advantages of publish/subscribe communication and mobile agents into a flexible and extensible distributed execution environment.

Faults and failures are inevitable in a distributed system built over a wide area network and they should be tolerated for continuing system transactions. We analyze effectiveness of the system on the failure of agent servers and disconnection of communication links. In these cases ABDES dynamically reconfigures the connections among agent servers in order to create paths that increase the performance of message routing. Fault tolerance by reconfiguring connections is integrated with the routing algorithm of ABDES and the method proposed here is highly scalable and robust to partial failures of agent servers in an agent based distributed event system.

References

1. Carzaniga, A., Rosenblum, O. and Wolf, A.: Design and Evaluation of a Wide Area Notification Service. *ACM Transactions on Computer Systems*, 3(19), (2001), 332–383.
2. Cugola, G., Nitto, E.D. and Fuggetta, A.: The JEDI Event-Based Infrastructure and its Applications to the Development of the OPSS WFMS. *IEEE Transactions on Software Engineering*, 27(9), (1998), 827–850.
3. Pietzuch, P. and Bacon, J.: Hermes: A Distributed Event-Based Middleware Architecture, In *Proceedings of the 1st International Workshop on Distributed Event-Based Systems (DEBS'02)*, (2002), 611-618
4. Picco, G. P., Cugola, G. and Murphy., A. L.: Efficient Content-Based Event Dispatching in the Presence of Topological Reconfiguration, *23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03)*, (2003), USA, 234–243.
5. Baldoni, R., Beraldi, R., Querzoni, L. and Virgillito, A.: A Self-Organizing Crash-Resilient Topology Management System for Content-Based Publish/Subscribe, *International Workshop on Distributed Event-Based Systems (DEBS '04)*, Edinburgh, Scotland, UK, (2004).
6. Xu, Z. and Srimani, P. K.: Self-Stabilizing Publish/Subscribe Protocol for P2P Networks, *Lecture Notes in Computer Science*, Springer-Verlag, Volume 3741; *Distributed Computing - IWDC 2005: 7th International Workshop*, Kharagpore, India, December 27-30, (2005), 129-140
7. Buchmann, A. et al, *DREAM: Distributed Reliable Event-based Application Management*, Springer, (2004), 319-350
8. Cugola, G., Frey, D., Murphy, A. L. and Picco, G.P.: Minimizing the Reconfiguration Overhead in Content-Based Publish-Subscribe, In *Proceedings of the 19th ACM Symposium on Applied Computing (SAC)*, Cyprus, (2004), 1134-1140.
9. V.S. Sunderam et al.: *Publish/Subscribe Systems on Node and Link Error Prone Mobile Environments*, ICCS 2005, Springer-Verlag, LNCS 3515, (2005), 576 – 584.
10. Sahingoz, O. K., and Erdogan N.: Agent-Based Distributed Event System, *Proceedings of 30th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2004)*, Czech Republic, (2004), 144-153.
11. Sahingoz, O. K., and Erdogan N.: Dispatching Mechanism of an Agent-Based Distributed Event System, LNCS, Springer-Verlag, Vol.3036 ICCS 2004, (2004), 184-191.

Link Speed Estimation and Incident Detection Using Clustering and Neuro-fuzzy Methods

Seung-Heon Lee, M. Viswanathan, and Young-Kyu Yang

College of Software, Kyungwon University, Bokjeong-Dong, Sujung-Gu,
Seongnam-Si, Gyeonggi-do, South Korea 461-701
shleejj@gmail.com, {murl, ykyang}@kyungwon.ac.kr

Abstract. The primary issues in the development of advanced traveler information systems (ATIS) within the intelligent transportation systems (ITS) framework are the optimal estimation of freeway travel time and incident detection with reasonable accuracy. Typically ATIS aims to provide route guidance based on the traveler's requirements using the information gathered from various sources such as loop detectors and probe vehicles. Until recent times traffic information was collected from mostly stationary devices and analyzed. In this research paper we consider data acquired from primarily GPS-based sources. The aim of the research is a comprehensive analysis of collected information from GPS sources using the fuzzy c-means algorithm (FCM) which provides the estimation of link speed. The modified FCM is used to extract patterns from the traffic data collected from a busy network of downtown streets. The link speed estimation is performed using smoothing techniques. Finally we apply the neuro-fuzzy algorithm to the task of incident detection from the traffic patterns.

1 Introduction

In recent years the growth in automobiles sales and the related economic and environmental factors have increased the pressure on infrastructure development. However due to the limits on infrastructure growth efficient and intelligent utilization of the existing transportation infrastructure has become most significant. Intelligent transportation systems are therefore needed to provide a high level of automation in advanced traffic management systems (ATMS) and advanced traveler information systems (ATIS). The accurate estimation of the traffic information is of utmost importance for the drivers as well as the traffic information service provider.

Until recent years, traffic data was collected by the fixed detectors such as loop detector, CCTV and beacon, etc. Many researchers have endeavored to develop reliable travel time forecasting and incident detection models using various methods including historical profile approaches, time series models, neural networks, nonparametric regression models, traffic simulation models, and dynamic traffic assignment (DTA) models [2]. The methods of incident algorithms can be grouped as algorithms based on pattern recognition and statistical methods, algorithms using traffic models and theory, and other hybrid state of the art incident detection methods based on neural networks and fuzzy methods. Due to the recent trend in the wide-spread use of GPS

receivers in car navigation systems, traffic information can be easily collected by the probe cars equipped with GPS receivers. Many traffic information providers are offering link speeds estimated solely by GPS data due to the low cost and data availability in real-time. This thus requires developing novel algorithms for extracting traffic patterns, different from the algorithms used for the conventional fixed detectors.

The aim of research is to propose and test a new algorithm for calculating the travel speed and incident detection for the collected information from the probe cars equipped with GPS receivers. For this aim, we automatically cluster the GPS data into five levels of speed layers which are very slow, slow, middle, high and very high speed using Fuzzy C-Means (FCM) and apply these layers of speed patterns to predict the link travel speed and incident detection.

2 Existing Approaches in Link Speed and Incident Detection

Typically many conventional methods use statistical and simple neural network algorithms to calculate the link travel speed. The data set collected by the fixed detection devices are speed, road occupancy rate, and the amount of traffic flow and available on for the fixed locations where the devices are installed. But, acquired data from GPS probe cars are only speeds at specified time and location. And therefore traffic information from GPS devices requires novel methods and algorithms for analysis.

Some examples of traffic information modeling include the ADVANCE project [9] which estimated the link travel time for expressway and arterial roads using traffic volume and occupancy ratio from detector data. Sisiopiku et al.[10] present a study of correlation between travel time and traffic volume and occupancy rate for estimation of link travel time. They use the NETSIM model for link travel time estimation and analyzed the relation between traffic volume and travel time as well as occupancy and travel time using Fuzzy and estimated link travel time. Nelson et al. [11] used traffic volume and occupancy rate as input data and estimated travel time using fuzzy-neural system studied integration of neural network with fuzzy rule with empirical results that suggested that the integration improved the performance. Some well-known pattern recognition algorithms include the California algorithm [5][7][8] that detects incidents using occupancy rate at specified times, APID algorithm [6][7][8] that compares occupancy between two detector areas, and wave analysis, PATREG, and the Monica algorithm. Statistical algorithms include SND and Bayesian algorithms [7] that use probability of incident occurrence through blocked crossroads of downstream traffic. Algorithms using traffic model and theory are Dynamic Algorithms that assumed sudden change in the freeway system to follow predictable patterns due to the integrated relation between speed-density and traffic-density in detection analysis. Additionally based on Catastrophe Theory, the (modified) McMaster Algorithm uses traffic theory as explanation of "Jump" of speed in the operation of freeway that has little change of occupancy. The final group of hybrid algorithms includes Fuzzy Set algorithms, Neural Network Algorithms, Probabilistic Neural Network and Automatic Vehicle Identification Systems [5][7]. The algorithms as mentioned above suffer from issues such as excessive training time for data analysis and the intervention of professionals for generating useful patterns. This research primarily attempts to adapt Fuzzy C-Means algorithm for solving these issues.

3 Design of Link Speed and Incident Detection Estimation System

The primary motivation is to generate patterns for calculating the link speed using an adapted FCM algorithm. The data collection interval is of 5 minute duration and the probe cars are GPS-equipped taxis. The data thus acquired includes patterns such as stopping and running, slow and fast running and searching-running. Due to the various traveling patterns, calculation using time mean speed has many gaps in comparison with actual speed. Also, it is difficult to detect incidents since the taxi probe cars have frequent lower speeds. Therefore, this research proposes a new speed calculation method for optimal estimation of the travel speed and robust incident detection. Figure 1 shows structure of system for link travel speed estimation and incident detection.

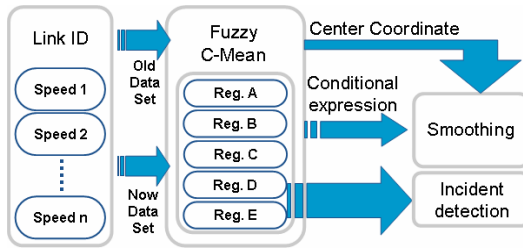


Fig. 1. Structure of system for calculating link speed and incident detection

Since urban traffic patterns have high variation extensive calculations are required and it is challenging to integrate all links. Therefore, in this research, a method using FCM is applied to analyze traffic patterns of each link and reduce calculation volume when generating patterns through training. For calculation of vehicle speed V using GPS we divide length of link by time using pass time of previous node and current node.

$$V = \text{Length of Link} / (T_{nL} - T_{pL}) \tag{1}$$

Using calculated link speed, we classify data into same time of interval of 5 minutes. Classified data is calculated three pattern (high, middle, low speed) using FCM. FCM recognizes spherical clouds of points in p-dimensional space. Having a finite set of objects $X=\{x_1, \dots, x_n\}$ and the number of cluster centers c to be calculated, the assignment of the n objects to the c clusters is represented by the proximity matrix $U=[U_{ik}]$. With $k=1, \dots, n$ and $i=1, \dots, c$, $u_{ik} \in [0,1]$ expressing the fuzzy proximity or affiliation of object x_k to cluster center v_i . Matrix U is used to calculate region of speed and v_i is used to calculate link speed. After generating the cluster we obtain the values of the cluster center and min-max value of cluster. Min-max values are used to decide boundary of speed layer from measured real-time GPS data for calculation of link speed and center values are used in conversion from low-speed to high speed.

Using matrix U , we make conditional expression for pattern of speed. Each cluster has maximum value, minimum value and center value for the speed. These values are used for calculating the link speed using the real time collection speed. V_T is time mean speed and can be calculated by (2). V_T is difficult to use for the link speed

production which uses GPS because volume of collected data using GPS is small until recently. Also, this equation has tendency to ignore high speed data and calculates to speed of middle inside putting first. Therefore, we propose a new speed production method for GPS data.

$$V_T = \frac{1}{N} \sum_{i=1}^N V_i \tag{2}$$

$$V_T = \frac{1}{N} \left(\sum_{j=0}^n V_{i1} + \sum_{k=0}^m (\alpha V_{i2} + (1 - \alpha)v_i) \right) \tag{3}$$

As v_i is center value of cluster which is included maximum speed, V_{i1} is all of speed which is included cluster. And V_{i2} is mean low speed that is little higher than center value of cluster. α are threshold values of speed for making near in high-speed layer. We calculate near maximum speed according to allow many weights to high speed more than low speed. In the above case, maximum speed include in one of pattern. But, FCM is able to get speed which is not include in speed domain U because it is not reflect on region which is not included speed data. Therefore, we calculate new standard speed non cluster v_i using (4). It is calculated by providing weight on center of upper and lower cluster of collected maximum speed. w is weight for connection between upper cluster and lower cluster and has a region from 0 to 1. Weight is calculated using (5).

$$v_i^{non\ cluster} = w \bullet v_i^{upper\ cluster} + (1 - w) \bullet v_i^{lower\ cluster} \tag{4}$$

$$w = N_{lower} / (N_{upper} + N_{lower}) \tag{5}$$

N which is used to calculate weight is number of objects including upper and lower cluster. A cluster with many objects has higher weight in relation to others. Once the weight is computed, the base speed for non-pattern area and link speed can be computed using (3) and (4). The estimated link speed is usually larger than the mean speed because the highest speed data of the link is used for the calculating the link speed. Finally, if estimated speed is higher than the speed limit, it is not usable due to the traffic regulations. Therefore, we will offer users the regulation speed in those cases.

The other aim of this paper is to adapt the ANFIS algorithm to detect incidents in a scenario where the available input data consists of speed measurements based on moving probe vehicles. For incident detection, input values are center values that were generated from our FCM algorithm, collected speed and previous mean speed. Work of first measure that whether or not inclusion about distribution of collected speed in low speed layer. At this time, in case of including all of collected speed in region D-E, our system regard as incident and decide to incident using collected speed, previous mean speed and cluster value of now and previous through ANFIS. The fuzzy membership function that used in ANFIS is constructed generating 36 rules for incident detection. After calculating the speed and pattern, we used this data as inputs and detect incidents through ANFIS. Figure 2 presents some of the fuzzy rules generated.

```

If (AVGSPD is SS) and (PDSPD is SS) and (PPATTERN is VLS) then (RESULT is INCIDENT)
If (AVGSPD is SS) and (PDSPD is MS) and (PPATTERN is VLS) then (RESULT is INCIDENT)
If (AVGSPD is SS) and (PDSPD is LS) and (PPATTERN is VLS) then (RESULT is INCIDENT)
.....
If (AVGSPD is SS) and (PDSPD is LS) and (PPATTERN is VLS) and (NPATTERN is LS) then (RESULT is INCIDENT)
If (AVGSPD is SS) and (PDSPD is MS) and (PPATTERN is VLS) and (NPATTERN is LS) then (RESULT is INCIDENT)
If (AVGSPD is SS) and (PDSPD is SS) and (PPATTERN is VLS) and (NPATTERN is LS) then (RESULT is INCIDENT)
    
```

Fig. 2. Fuzzy rules for incident detection

First, the four input values which include speeds of link and center values of cluster current and previous in each observation are given to the system. For each value the membership degree is calculated by the fuzzy membership function and these values are passed into the neural network according to the fuzzy rules. The neural network multiplies weights to value of all rules and then the resulting values after addition are sent to the output layer. These values undergo de-fuzzification by the fuzzy membership function and the final output value is acquired. The structure of the adapted ANFIS algorithm is shown in figure 3. Once the error rate of the model reaches 0.01 or the learning has gone through 600 epochs, the process is terminated.

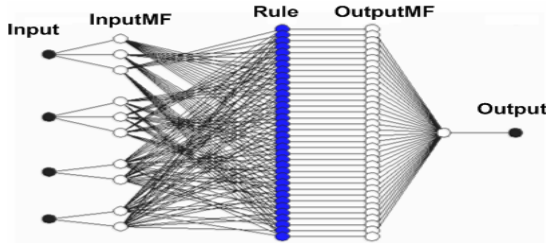


Fig. 3. Anfis model structure

4 Empirical Framework and Analysis

In the creation of speed pattern, we employ taxi-based probe cars for data collection. For link speed analysis and data generation the GPS-based probe cars acquire data for a 1 hour period between 9-10 am. The data thus collected is studied for patterns using FCM. Table 1 shows center value of each cluster after analysis using FCM at 5 minute interval until during the 1 hour period. We generate 5 clusters and get data about center and min-max values of cluster for all days excepting weekend. Table 2 presents results of comparing our algorithm with time mean speed and actual speed. Our algorithm is often closer to the actual speed than the time mean speed. The results from the following table suggest that our algorithm has less variance in approximating the actual speed in comparison to time mean speed. Through performance testing, the average error rate of speed between our proposed algorithm and time mean speed are 4.36km and 9.20km respectively.

Table 1. Center of calculated speed pattern using Fuzzy C-Mean

Speed	Very Low	Low	Middle	High	Very High
09:00-09:05	4	8	10	15	23
09:05-09:10	4	9	13	20	28
09:10-09:15	4	8	11	17	29
09:15-09:20	6	10	15	22	29
09:20-09:25	6	10	16	24	38
09:25-09:30	4	8	12	18	32
09:30-09:35	5	10	15	30	60
09:35-09:40	5	10	13	20	39
09:40-09:45	5	11	19	29	51
09:45-09:50	5	10	17	29	49
09:50-09:55	6	12	20	30	48
09:55-10:00	6	12	23	38	62

Table 2. Comparison of calculated speed using mean, test probe car and our algorithm

5 interval	Link 1			Link 2		
	mean	actual	new	mean	actual	new
09:00-09:05	47	43	54	15	31	20
09:05-09:10	36	41	43	24	31	31
09:10-09:15	48	45	48	19	29	25
09:15-09:20	24	43	40	20	26	27
09:20-09:25	33	37	41	22	24	29
09:25-09:30	34	38	44	15	27	25
09:30-09:35	42	42	48	10	25	12
09:35-09:40	25	48	42	26	23	27
09:40-09:45	37	50	49	12	24	18
09:45-09:50	36	51	43	14	29	26
09:50-09:55	43	47	45	20	33	33
09:55-10:00	42	45	42	23	37	36

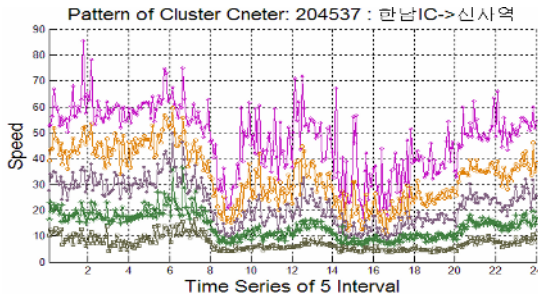


Fig. 4. Speed variation over 1 month between 0:00 am to 12:00 pm

During the office-going hours and closing times from 2:00 PM, there is sudden lowering of speed and at low-peak hours rising speeds are observed as in figure 4. Furthermore, speed distribution varies across different links due to the state of road and environmental factors. Figure 4 shows variation of speed about the center value of each cluster.

Performance of the proposed model about incident detection is evaluated by using the detection rate and false alarm rate of incident detection. Formula for the incident detection rate (DR) and false alarm rate (FAR) are as follows:

$$DR = t / (t + t'), \text{ FAR} = f / (t \times s) \tag{6}$$

t is number of real incident, t' is number of incident that were undetected, f is number of false alarms on incident detection and s is number of links.

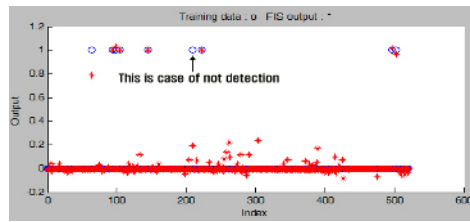


Fig. 5. Result of learning using ANFIS

Figure 5 shows result of learning by using speed and increase and decrease rate. Blue circle represents target value and red star the real output value. In figure5 successful incident detection in seen in 8 cases and misclassification in a single case when using a data set of 1200 input values. Based on the empirical analysis the average detection rate is over 83% and the false alarm rate is under 23%. Our algorithm has plausible performance when applied to the collected GPS data and is robust in the presence of limited data.

5 Conclusion

The speed variation of vehicles in urban streets poses serious challenges in link travel time estimation and incident detection. Additional factors like Road state, environmental factors and the integration of speed data from multiple links result in a high volume of computation. In attempting to develop an optimal link travel time algorithm we adapt the well-known FCM for clustering and link speed estimation from real-time GPS-based probe car data. A robust incident detection model was also developed using the ANFIS system in conjunction with the feedback from our FCM algorithm. The empirical analysis suggests that our approach offers plausible performance in the presence of limited data and high speed variance. Our approach also offers a reduction in the computational costs and robustness in the presence of link failures. While this research aims to enhance accuracy in estimation of link speed and incident detection further work is required in fusing data from multiple sources and integrating speed information from arterial roads and expressways.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program super-vised by the IITA (Institute of Information Technology Assessment).

References

1. Hoppner, F.: "A Contribution to Convergence Theory of Fuzzy c-Mean and Derivatives," IEEE Transactions of Fuzzy System, Vol. 11. NO 5, 2003
2. You, J., Kim, T.J.: "Development and evaluation of a hybrid travel time forecasting model," Transportation Research Part C: Emerging Technologies, Vol. 8. Issues 1-6, 2000, pp. 231-256
3. Bruce R. Hellinga, Liping Fu, "Reducing bias in probe-based arterial link travel time estimates," Transportation Research Part C: Emerging Technologies, Vol. 10. Issue 4, 2002, pp. 257-273
4. Jesús Lázaro, , Jagoba Arias, José L. Martín, Carlos Cuadrado, Armando Astarloa, "Implementation of a modified Fuzzy C-Means clustering algorithm for real-time applications," Microprocessors and Microsystems, Vol. 29. Issues 8-9, 2005, pp. 375-380
5. Dipti Srinivasan, Ruey Long Cheu, Young Peng Poh and Albert Kim Chwee Ng, "Development of an intelligent technique for traffic network incident detection", Engineering Applications of Artificial Intelligence, Vol. 13, Issue 3, 1 June 2000, pp. 311-322.
6. Dipti Srinivasan, Xin Jin and Ruey Long Cheu, "Adaptive neural network models for automatic incident detection on freeways", Neurocomputing, Vol. 64, March 2005, pp. 473-496.
7. Sheu, J.B.: "A fuzzy clustering-based approach to automatic freeway incident detection and characterization", Fuzzy Sets and Systems, Vol. 128, Issue 3, 16 June 2002, pp. 377-388.
8. Jih-Biing Sheu, "A sequential detection approach to real-time freeway incident detection and characterization", European Journal of Operational Research, Vol. 157, Issue 2, 1 September 2004, pp. 471-485.
9. Boyce, D., Roupail, N., Kirson, "A Estimation and measurement of link travel times in ADVANCE project," Proceedings of Vehicle Navigation and Information Systems Conference, IEEE, 1993, pp.62-66.
10. Sisiopiku, V., Roupail, N.: "Exploratory Analysis of the Correlations Between Arterial link Travel Times and Detector Data from Simulation and Field Studies," ADVANCE Working Paper No. 25, Oct. 1993.
11. Nelson, P. and Palacharla, P.: "A neural network model for data fusion in ADVANCE," in Proceedings of Pacific Rim Conference, Seattle, Wash., 1993, pp. 237-43.

A Consensus-Based Multi-agent Approach for Information Retrieval in Internet

Ngoc Thanh Nguyen¹, Maria Ganzha², and Marcin Paprzycki³

¹Institute of Information Science and Engineering, Wroclaw University of Technology, Poland
thanh@pwr.wroc.pl

²Department of Management, Elbląg University of Humanities and Economy, Poland
ganzha@euh-e.edu.pl

³Computer Science Institute, Warsaw School of Social Psychology, Poland
marcin.paprzycki@swps.edu.pl

Abstract. This paper presents a consensus-based approach utilized within a multi-agent system which assists users in retrieving information from the Internet. In this system consensus methods are applied for reconciling inconsistencies among independent answers generated by agents (using different search engines) for a given query. Proposed agent system has been implemented and initial experimental results are presented.

1 Introduction

Information retrieval is one of tasks, which are the most often realized by computer users and a large number of methods, technologies and search engines have been proposed for aiding them in this task. However, most users utilize only a single search engine. They trust it that the obtained answer is relevant and complete. But an interesting question arises: with so many existing search engines – why to use only one? It is quite possible that different search engines provide “best” responses to different queries.

Consensus methods, based on the consensus theory that states that if the same task is entrusted to several experts (and their credibility is approximately the same) then their reconciled solution (the *consensus answer* or the *consensus* – we use these terms interchangeably in what follows) should be more credible than those generated by them individually, have been proved to be useful in dealing with data originating from multiple sources [1], [4], [6]. Similarly, consensus based on responses generated by multiple search engines should be more relevant than each of individual responses. This fact motivates usage of multiple search engines in retrieving information from the Internet.

The initial predicted role of agent systems was in the area of information management [12]. In this context agent autonomy and communication have been particularly useful [5]. These features make them also natural candidates for realization of “consensus computing systems,” created with the goal of improving quality of information retrieval. At the same time, there are very few such systems in existence. Menczer [9] has designed and implemented *Myspiders*, a multi-agent system for information

discovery in the Internet and has shown that augmenting search engines with adaptive intelligent search agents can lead to significant competitive advantages. Another approach was realized in the www.metacrawler.com, which utilizes multiple search engines.

In this paper we present a novel agent system designed to assist information retrieval from the Internet. Our system differs from the one of Menczer in using consensus methods [10, 11] for resolving differences in response sets and using multiple agents for the retrieval task. The advantage of this approach is that agents use different search engines (among themselves and for subsequent queries). To create the final answer we use the consensus method to select URLs' that are ranked as minimally distant from those provided by other agents. The *consensus answer* is displayed to the user and, due to the way it was obtained, it is expected to be more complete and relevant. Furthermore, each agent has its knowledge base containing information about results of past searches and used to select the search engine to be utilized in subsequent searches. Let us now describe in more details the proposed approach, followed by the set of initial experimental results.

2 Multi-agent System Design

2.1 Motivation

The aim of the project is to create a consensus-based multi-agent system to aid users in information retrieval from the Internet. This approach enables to solve the following two problems often occurring in the information retrieval processes:

- *Low relevance of answers generated by a search engine* – caused by non-effective work of filters. As a consequence, many non-related pages (e.g. advertisements, but also pages loosely associated with the subject of the query) may be displayed. For instance, for the query „Wroclaw University of Technology” *Yahoo* as the most relevant gives <http://www.pwr.wroc.pl/~promocja/eng/main2.html> which can be considered a “correct” answer. However, *Google* classifies this URL on the 2nd position, while *Onet* does not include it within first 60 displayed URLs.

- *Displaying repeating URLs* – which are identical or very similar to each other. This is a burden for the user because he loses a lot of time to scroll many screens to find information of interest.

The proposed system exploits answers generated by multiple search engines. Utilizing the consensus algorithm, their optimal combination is determined and displayed to the user. Answer-sets (and thus search engines) are evaluated on the basis of their differences from the *consensus answer*. This information is fed-back to search agents, stored and utilized in subsequent queries to select the search engine to be used.

2.2 System Design

The general structure of the system is represented in Figure 1 as a UseCase diagram. Within the system we can see two types of agents: (1) Search Agent (SA) and (2) Manager Agent (MA). We assume that for each user there will be a single MA, while multiple SAs will be created to utilize multiple search engines. For the time

being we assume also that the number of agents is smaller than the total number of available search engines. As can be seen the SA obtains a query form the MA and after selecting the search engine (which is based on the information stored in the knowledge base (KDB) and coordinated by the MA) it queries it and prepares the report for the MA (this function involves removal of multiple URLs and selecting n best answers). On the basis of evaluation of its work, performed by the MA, the SA updates its KDB. The MA, on the other hand, receives a query from the user and sends it to the SAs. It oversees the process of search engine selection (each SA is to utilize a different one). Upon reception of the search results from all agents it determines the *consensus answer* which will be presented to the user [11]. However, before the answer is presented its *consistency* is established. If consistency is high (results were similar), then the results are used immediately to rank SAs (results close to the consensus are ranked higher than these that are not). If consistency is low (search engines did not agree on the answer set) then user is asked to pick relevant answers. These picks are then used to rank answers and provide feedback to SAs. If the user does not pick any answers (for whatever reason, as she is not forced to do so), no feedbacks is send.

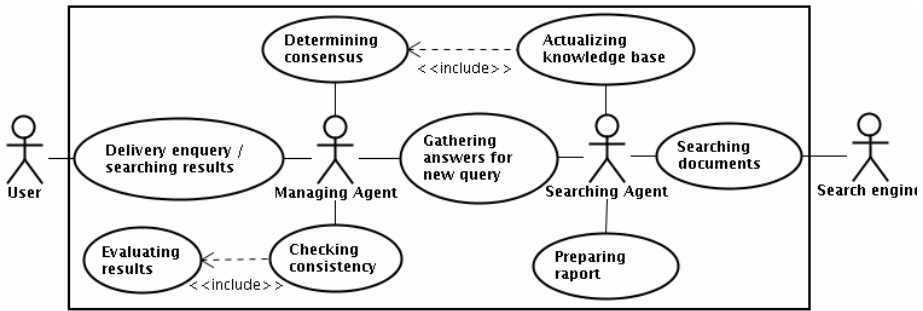


Fig. 1. System Use Case diagram

Thus far we have omitted a number of administrative functions that are involved in, among others, system start-up and system shut-down. These functions are: (1) Upon system start-up all necessary agents are created (first the MA, that creates a specific number of SAs) and the SAs load information about search engines (e.g. location, interface, etc.) from a shared database. (2) Each SA loads, from the KDB, individual information that is used in search engine selection. (3) Upon system shut-down each SA stores content of its KDB for use during the next system run.

Let us now assume that the system is running (all agents have been initialized and information about at least some searches is stored in each SA’s KDB). In Figure 2 we present a complete UML Action Diagram of interactions between the MA and an SA during servicing a user query. Let us note that the selection of the search engine involves both the information stored in the KDB and interactions with the MA. To prevent all SAs from converging on a single search engine we force each SA to use a different one to process each query. This process is completely asynchronous; e.g. the

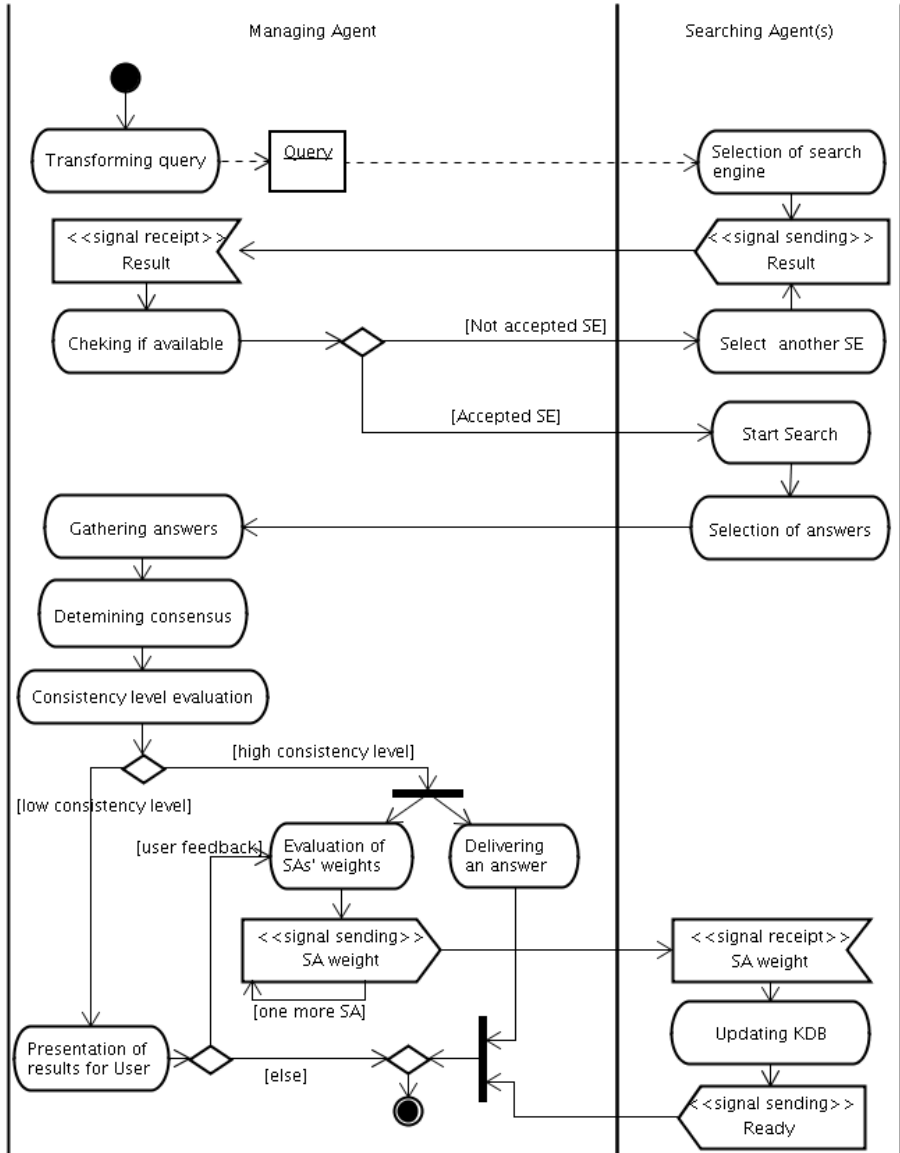


Fig. 2. Servicing user query – UML activity diagram

first SA to select *Google* and contact the MA will be able to use it, while the remaining SAs interested in *Googe* will have to select a different search engine. The MA oversees this process until all SAs have selected a different search engine. To simplify this process we have decided to utilize smaller number of agents than the number of available search engines and allow them to select randomly the next search engine (in the case when they have to use one that is not their favorite).

2.3 Main Algorithms Used in the System

Let us now describe key computational processes that take place in the system. First the MA finds the *consensus answer* using a version of the *branch-and-bound* algorithm [2] (we assume that there are m SAs and they report n results each):

Input: *results* provided by m SAs – $r^{(1)}, r^{(2)}, \dots, r^{(m)}$ each in the form $r^{(i)} = \langle d^{(i)}_1, d^{(i)}_2, \dots, d^{(i)}_n \rangle$ where $d^{(i)}_1, d^{(i)}_2, \dots, d^{(i)}_n$ are URLs representing documents for $i = 1, 2, \dots, n$.

Output: *consensus answer* $C = \langle d_1, d_2, \dots, d_n \rangle$

BEGIN

1. create set D of all documents from all *results* (without repetitions)
2. for each $d \in D$
 - create vector $\langle t_1, t_2, \dots, t_n \rangle$ where t_j is the position on which d appears in $r^{(i)}$; if d does not appear in $r^{(i)}$ then set t_j as the length of the longest ranking increased by 1
 - calculate average $t(d)$ of values t_1, t_2, \dots, t_n
3. *consensus answer* C is obtained by ordering elements of D according to values $t(d)$
4. since size of D can be large one can utilize only a few top results.

END.

The resulting *consensus answer* represents a combined view on what the search engines believe the final answer should be. Unfortunately such a final answer is not always reliable. We can say that it is reliable only when all search engines have a very similar view as to what the final answer should be. To be able to establish how similar obtained answers are we calculate the *consistency of the consensus*. This is achieved by applying the following algorithm (see also [11]):

Input: Set X consisting of n binary matrices $A^{(k)}$ (for $k=1, \dots, n$) of size $m \times m$ representing individual rankings (result sets)

Output: YES if consensus is consistent; NO otherwise

BEGIN

1. For all rankings calculate consensus C

$$2. \text{ Calculate } \hat{d}(X) = \frac{\sum_{x,y \in X} d(x,y)}{m(m+1)} ;$$

$$3. \text{ Calculate } \hat{d}_{\min}(X) = \frac{\sum_{y \in X} d(C,y)}{m} ;$$

4. If $\hat{d}_x(X) \geq \hat{d}_{\min}(X)$ then YES; NO otherwise

END.

In the case when the *consistency of the consensus* is low (below a threshold; and the above algorithm responds with a NO) results are shown to the user and she is requested to provide explicit feedback by selecting responses that she finds to be

relevant to the query. These responses are then used to rank the search engines. This is done in the similar way to what happens when consistency is high enough. Then the following operations take place (C is the *consensus answer* calculated as above):

- + distance $d(C, A^{(j)})$ for $j = 1, \dots, m$ is calculated
- + such k that $d(C, A^{(k)})$ is minimal is found
- + weight for agent k is then assumed to be $W[k] = 100\%$;
- + for each remaining agents SA its weight is equal to:

$$W[j] := \frac{\bar{dl}(p_i \cap q_j)}{\max\{dl(p_i), dl(q_j)\}} 100\%$$

where $\bar{dl}(p_i \cap q_j)$ is the length of common part of strings p_i and q_j starting from their beginning, and $j = 1, \dots, m, j \neq k$.

Weights $W[j]$ are then sent back to each SA and will be used to update its KDB and thus utilized when the search engine for the next query is to be selected.

The following other algorithms are utilized in the system (their detailed descriptions are given in report [3]):

- Algorithm for eliminating repeating URLs,
- Algorithm for transferring a binary matrix into a ranking,
- Algorithm for calculating distances between rankings,
- Three algorithms for calculating three types of distances between queries.

Let us now describe in some more detail finding distances between queries. User can express queries using logical forms with conjunctions “ \wedge ” and “ \neg ”, in the form $(a \wedge b) \wedge (\neg c)$, where a, b and c are terms. For example,

$$(consensus \wedge conflict) \wedge (\neg voting).$$

This query is extracted from the interface by the MA and passed to the SA which reformulates it for the selected search engine (e.g. interia.pl) as an HTTP statement:

<http://szukaj.interia.pl/id/query?mss=search&q=consensus+conflict+~voting&eng=dc>.

Before this step takes place, the SA has to select which search engine it would like to use. In its KDB it stores past queries and success achieved in responding to them using various search engines. New queries are compared to the old ones, among others, using the following algorithm (see also [10, 11]):

Input: queries p and q in the form $p=(p_1 \vee p_2) \wedge (\neg p_3)$ and $q=(q_1 \vee q_2) \wedge (\neg q_3)$,

Output: distance $d(p, q)$.

BEGIN

1. Calculate $d(p_i, q_j) = \frac{\bar{dl}(p_i \cap q_j)}{\max\{dl(p_i), dl(q_j)\}}$ for $i, j=1, 2$, where $dl(x)$ is the length of string x ;
2. Calculate $d = \max\{d(p_i, q_j) \text{ for } i, j=1, 2\}$;
3. Calculate $d' = d(p_3, q_3)$ using the pattern in step 1;
4. Calculate $d(p, q) = (1/2)(d + d')$

END

3 Initial Experimental Results

The proposed system has been implemented using Aglets agent environment [7, 8]. Specifically, we used JDK 1.1.6, Swing 1.0.1, JBCL 2.0 and IBM Aglets 1.1b3. For response sets generated by each search engine we have removed URL's that were more than 80% similar, assuming that they point to the same answer, and left only one. Then we have used 30 best responses. In experiments described below 20 search engines and 6 SAs were used. Finally, an SA was allowed to try to assign a search engine to the query only if its accuracy in the KDB was above 50%. Otherwise it was forced to select a random search engine.

We performed two tests. At the beginning of each test KDBs of all SAs were empty. The first test involved query „głosowanie \wedge wybory” („voting \wedge election”, in Polish). Table 1 contains the information about the search engines used by SAs and resulting weights informing about the similarity degree between the URLs' rankings provided by SA agents and the *consensus answer*. In all samples the consistency level of response sets was high enough to not to involve the user.

Table 1. Results of 6 samples for query: „głosowanie \wedge wybory”

Sample	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6
Sample I.	Onet 84%	Ahoj 100%	ICM 65%	Hoga 51%	Interia 56%	Google 43%
Sample II	Onet 51%	Ahoj 100%	ICM 21%	Hoga 25%	Interia 27%	WP 36%
Sample III	Onet 100%	Ahoj 52%	Arena 21%	WP 28%	Google 49%	Poland 45%
Sample IV	Onet 100%	Ahoj 52%	Hoga 21%	ICM 18%	Poland 19%	Google 33%
Sample V	Onet 100%	Ahoj 52%	ICM 21%	Hoga 18%	Interia 19%	WP 44%
Sample VI	Onet 100%	Ahoj 72%	Poland 56%	ICM 25%	Arena 32%	Hoga 14%

From Table 1 follows that the best search engines turned out to be Onet and Ahoj (both search engines focused on Polish web sites). We can also notice that in the first retrieval majority of weights for search engines are high enough (more than 50%). This means that rankings generated by SAs are consistent to the large degree. Therefore, in the second sample agents 1-5 used the same search engines while agent 6 had to draw a new one. This was the reason of change of weights to the disadvantage of agents 3, 4 and 5. In the next samples only agents 1 and 2 have good enough accuracy and can use the same search engines (Onet and Ahoj) while other agents had to draw. It is worth noting that because the query was in Polish, therefore agents draw only from Polish search engines. After checking by an expert it turned out that all responses were relevant.

In the second test the system performed retrieval for 50 different queries related to “holidays.” On the basis of results obtained in these tests we can draw the following conclusions:

- Utilization of KDBs caused SAs to less and less often draw search engines, rather these selections have been determined on the basis of previous queries.
- Growth of information stored in KDBs resulted in results being more and more relevant (e.g. number of „pushy” URLs has been smaller and smaller). In the final rankings for the last query „pushy” URLs have not occurred.

4 Conclusions

While utilization of several search engines for the same query is not novel, method for reconciling the results presented here is. Its advantage is that it does not need the information about the user (her preferences, profiles etc.). It works on the basis of an assumption that if several experts (search engines) solve the same problem, then the reconciled solution should be more credible than those proposed by individual experts. Future work should concern using advanced inconsistency measures to evaluating conflict situations and making decision for consensus determination [11]. We also plan to utilize the proposed approach for content collection within the agent-based travel support system described in [13].

References

1. Barthelemy, J.P., Janowitz M.F.: A Formal Theory of Consensus. *SIAM J. Discrete Math.* **4** (1991) 305-322
2. Barthelemy, J.P., Guenoche, A., Hudry, O.: Median linear orders: Heuristics and a branch and bound algorithm. *European Journal of Operational Research* **42** (1989) 313-325
3. Błazowski, A, Nguyen, N.T.: AGWI- Multi-agent System Aiding Information Retrieval in Internet. In: Proceedings of SOFSEM 2005. *Lecture Notes in Computer Science* **3381** (2005) 399-403.
4. Day, W.H.E.: Consensus Methods as Tools for Data Analysis. In: Bock, H.H. (ed.): *Classification and Related Methods for Data Analysis*. North-Holland (1988) 312-324
5. Ferber, J.: *Multi-Agent Systems*. Addison Wesley, New York (1999)
6. Katarzyniak R.P., Pieczynska-Kuchtiak A.: A consensus based algorithm for grounding belief formulas in internally stored perceptions. *Neural Network World* **5** (2002) 461-472
7. Lange, D., Oshima, M.: *Programming and Developing Java™ Mobile Agents with Aglets*, Longman (1998)
8. Lange, D., Oshima, M.: *Java agent API: Programming and deploying aglets with Java*. Aglets Web page: <http://www.ibm.co.jp/trl/projects/aglets/>
9. Menczer, F.: Complementing Search Engines with Online Web Mining Agents. *Decision Support Systems* **35** (2003) 195-212
10. Nguyen, N.T.: Consensus System for Solving Conflicts in Distributed Systems. *Journal of Information Sciences* **147** (2002) 91-122
11. Nguyen, N.T., Malowiecki, M.: Consistency Measures for Conflict Profiles. *LNCS Transactions on Rough Sets* **1** (2004) 169-186.
12. Maes P.: Agents that Reduce Work and Information Overload. *Communications of the ACM*, **37**, 7 (1994) 31-40
13. Ganzha M., Gawinecki M., Paprzycki M., Gąsiorowski R., Hyska W., Pisarek S.: *Utilizing Semantic Web and Software Agents in a Travel Support System*, Idea Publishing (to appear)

An Adaptive Fuzzy kNN Text Classifier

Wenqian Shang¹, Houkuan Huang¹, Haibin Zhu², Yongmin Lin¹,
Youli Qu¹, and Hongbin Dong¹

¹ School of Computer and Information Technology, Beijing Jiaotong University, 100044,
China

shangwenqian@hotmail.com

² Senior Member, IEEE, Dept. of Computer Science, Nipissing University, North Bay,
ON P1B 8L7, Canada

haibinz@nipissingu.ca

Abstract. In recent years, kNN algorithm is paid attention by many researchers and is proved one of the best text categorization algorithms. Text categorization is according to training set which is assigned class label to decide a new document which is not assigned class label belongs to some kind of document. Until now, kNN algorithm has still some issues to need to study further. Such as: improvement of decision rule; selection of k value; selection of dimensions (i.e. feature set selection); problems of multiclass text categorization; the algorithm's executive efficiency (time and space) etc. In this paper, we mainly focus on improvement of decision rule and dimension selection. We design an adaptive fuzzy kNN text classifier. Here the adaptive indicate the adaptive of dimension selection. The experiment results show that our algorithm is effective and feasible.

1 Introduction

With the development of the web, large numbers of documents are available on the Internet. Automatic text categorization becomes more and more important for dealing with massive data. It becomes a key technology to deal with and organize large numbers of documents. More and more methods based on statistical theory and machine learning has been applied to text categorization in recent years. For example, k-nearest neighbor (kNN)[1]-[4], Naive Bayes[5][6], Decision Tree[7][8], Support Vector Machines (SVM)[9], Linear least squares fit[10], neural network[11][12], SWAP-1, Rocchio and so on. Among these algorithms, kNN algorithm is studied by many researchers and is proved one of the best text categorization algorithms.

In recent years, many researchers study the improvement of kNN when the class distribution is uneven. They mainly focus on the improvement of decision function to resolve the problem of uneven class distribution, such as [13]-[17]. Our algorithm is based on the improvement of decision function too, but our improvement is very different from theirs. We adopt the theory of fuzzy sets, through analyzing the relationship among distance, similarity and membership function, according to the similarity, design a new weighted factor, at the same time we study the effect of dimension selection to

categorization performance. We design a formula of dimension selection. The experiment shows that our improvement is feasible.

2 The Classical kNN Algorithm Based on SWF Rule

At present, there are two main decision rules in kNN algorithm, that is, the discrete value rule DVF (Discrete-Valued Function) and the weighted similarity rule SWF (Similarity-Weighted Function). The most widely used is the SWF rule. This paper mainly focuses on this rule. The kNN algorithm based on SWF rule can be described as follows:

The system searches k documents (called neighbors) that have the maximal similarity (cosine similarity) in training sets. According to what classes these neighbors are affiliated with, it grades the test document's candidate classes. The similarity between the neighbor document and the test document is taken as this class weight of neighbor documents. The decision function can be defined as follows:

$$\mu_j(X) = \sum_{i=1}^k \mu_j(X_i) sim(X, X_i) \tag{1}$$

Where $\mu_j(X_i) \in \{0,1\}$ shows whether X_i belongs to ω_j ($\mu_j(X_i) = 1$ is True) or not ($\mu_j(X_i) = 0$ is False); where ω_j is the sort of document class; $sim(X, X_i)$ denotes the similarity between training document and test document. Then the decision rule is: If $\mu_j(X) = \max_i \mu_i(X)$, then $X \in \omega_j$.

3 The Improved kNN Decision Rule

In classical kNN algorithm, there is an obvious problem: when the density of training data is uneven it may decrease the precision of classification if we only consider the sequence of first k nearest neighbors but do not consider the differences of distances. To solve this problem, we adopt the theory of fuzzy sets, constructing a new membership function based on document similarities as follows:

$$\mu_j(X) = \frac{\sum_{i=1}^k \mu_j(X_i) sim(X, X_i) \frac{1}{(1 - sim(X, X_i))^{2/(b-1)}}}{\sum_{i=1}^k \frac{1}{(1 - sim(X, X_i))^{2/(b-1)}}} \tag{2}$$

Where $j=1, 2, \dots, c$, $\mu_j(X_i) sim(X, X_i)$ is the membership of known sample X to class j. If sample X belongs to class j then the value is 1, otherwise 0. From this formula, we can see that in reality the membership is using the different distance of every neighbor to the candidate classifying sample to weigh its effect. Parameter b is

used to adjust the degree of a distance weight. From paper [6] we can know that the best value field of b is between 1.5 and 2.5, always using 2, in this paper we take the value 2. Then fuzzy k -nearest neighbors' decision rule is: If $\mu_j(X) = \max_i \mu_i(X)$, then $X \in \omega_j$.

Why we amend formula (1) to formula (2), the reasons mainly are:

1) Similarity has relation to distance

A distance function or a similarity function is a general measurement of pattern recognition. In topology, a distance can be defined as [18]: suppose the space is Ω , x and y are arbitrary two points in this space. Mapping $d(x, y) : \Omega \times \Omega \rightarrow R^+$ is called the two points' distance, if the mapping satisfies three conditions as follows:

- (1) $d(x, x) = 0, \forall x \in \Omega$;
- (2) $d(x, y) = d(y, x), \forall x, y \in \Omega$;
- (3) $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in \Omega$.

If the $d(x, y)$'s value field is $[0, 1]$, i.e., $d(x, y) : \Omega \times \Omega \rightarrow [0, 1]$, then $d(x, y)$ is called unitary distance between x and y and is expressed by $d_0(x, y)$.

We can use a unitary distance $d_0(x, y)$ to express similarity $sim(x, y)$. $Sim(x, y)$ can be defined as follows [19]: $f : d_0(x, y) \rightarrow sim(x, y) \in [0, 1]$, where f must satisfy the following conditions:

- If $d_0(x_1, y_1) > d_0(x_2, y_2)$, then $sim(x_1, y_1) \leq sim(x_2, y_2)$;
- If $d_0(x_1, y_1) < d_0(x_2, y_2)$, then $sim(x_1, y_1) \geq sim(x_2, y_2)$;
- If $d_0(x_1, y_1) = d_0(x_2, y_2)$, then $sim(x_1, y_1) = sim(x_2, y_2)$.

Through the above definition, we can educe the function relationship between a distance and a similarity from the definition of distance, which is given independently; whereas we can educe another transfer relationship from the definition of distance, which is given by similarity. This is the internal relationship between a distance and a similarity. The relationship between a unitary distance $d_0(x, y)$ and a similarity $sim(x, y)$ of arbitrary two points in feature space Ω can be described using complementary function $sim(x, y) = 1 - d_0(x, y)$ [19].

2) The similarity has relation to membership

The value field of a similarity and a membership is $[0, 1]$. This is not an occasion but a result of consanguineous affiliation between them. Through the concept of existing fields, we can relate a similarity with a membership. The existent field can be described as follows: around a point x in feature space Ω forms a field, for an arbitrary point y in the space, the value of this field can be measured by the similarity $sim(x, y)$ that y towards x . Such a field is called an existing field. This field forms a fuzzy set

around the center point x , can be described as A . Any point in the field belongs to A . Its membership can be measured by the similarity between this point and the center point x . The nearer the distance is to x , the higher the possibility it belongs to A . Contrariwise it is not. The value of membership in the center point x is maximal, i.e., the constant value 1. Hence, from the meaning of existing field, around a point x in feature space Ω , there exists a potential fuzzy set A , the similarity which any point y for A in this set can be measured by the similarity between y and x , that is, $\mu_A(y) = sim(x, y)$ [19].

4 The Research of Dimension Selection

In text categorization, dimension is defined as the number of the feature words in VSM (Vector Space Model). For text documents, the dimension is always thousands upon thousands. Such high dimension is not permitted by a classifier. This is so called curse of dimensionality. Among this feature words, only a lot is useful for text categorization, many of them are noise words which hurt the performance of text categorization.

At present, there are many methods to reduce the dimension space. In text categorization, people always use feature selection method to reduce the dimension, such as Information Gain, Cross Entropy, Mutual Information, CHI, Weight Evidence of Text, Odds Ratio and so on. These methods can reduce the dimension space greatly. But there still over thousands feature words after feature selection. So how many dimensions to be selected are proper; how many dimensions to be selected are to make the categorization performance best.

After we study other authors' experiments in kNN and its variants and the research of our own experiment, we find that when the classes and number of documents in training set are certain we can approximately make sure the dimensions of selection. It follows the formula as follows:

$$\text{dimensions} = \left\lfloor \frac{\lfloor \log(\text{num}(\max(\text{class}))) \rfloor}{\lceil \ln(\text{num}(\min(\text{class}))) \rceil} \right\rfloor \times 1000 \quad (3)$$

Where $\text{num}(\max(\text{class}))$ is the document numbers of the maximum class in training set, $\text{num}(\min(\text{class}))$ is the document numbers of the minimum class in training set. If

$\left\lfloor \frac{\lfloor \log(\text{num}(\max(\text{class}))) \rfloor}{\lceil \ln(\text{num}(\min(\text{class}))) \rceil} \right\rfloor$ is less than 1, we consider it as 1.

5 Experiment

5.1 The Datasets

In this paper, we use two datasets to validate our algorithm. One dataset comes from the International Database Center, Dept. of Computing and Information Technology,

Fudan University, China. The other dataset is Reuters-21578. We select 15 classes among 90 classes. The distribution of the class is uneven.

5.2 Experiment Result and Analysis

In experiment of Fudan's dataset, the feature dimension is 2000; the step of k is 5. In experiment of Reuters-21578, the feature dimension is 1000; the step of k is 5 too. The experiment result can be described as Table 1 and Table 2:

Table 1. The categorization performance in Chinese dataset when k is different

Value k	kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
10	81.972	79.907	83.802	82.346
15	81.571	79.326	83.441	81.882
20	81.077	78.397	82.564	80.720
25	80.717	77.468	82.547	80.372
30	80.307	76.887	81.876	79.443
35	80.061	76.423	81.627	78.978
40	78.458	74.100	81.016	78.165
45	77.885	74.100	80.612	77.700
50	77.657	72.822	80.232	77.120

Table 2. The categorization performance in Reuters-21578 dataset when k is different

Value k	kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
10	67.000	86.664	68.529	86.483
15	66.855	86.628	69.472	86.991
20	66.715	86.737	69.179	86.919
25	66.942	85.519	69.560	87.028
30	66.540	86.337	69.499	86.882
35	65.738	86.374	68.300	86.810
40	65.033	86.374	68.050	86.919
45	64.446	86.337	68.364	87.100
50	60.739	86.265	67.827	86.919

From Table 1 and Table 2, we can find that the improved kNN algorithm (fkNN) show better categorization performance than kNN no matter what the dataset is Chinese data or Reuters-21578. It has about 3% improvement in average performance. This proves that our improvement in decision rule is effective and feasible. This method solves the uneven problem of class distribution better.

Table 3. The categorization performance in Chinese dataset when dimension is different. Note that the highest accuracy is highlighted with bold font

dimension	kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
1000	80.207	76.655	82.133	79.791
2000	77.735	73.287	80.612	77.700
3000	78.219	73.287	81.931	79.210
4000	76.150	70.151	80.155	76.423
5000	76.396	70.267	79.306	74.913
6000	74.671	67.596	77.440	72.358
7000	69.258	60.163	74.595	67.712
8000	76.150	70.151	70.527	62.137

Table 4. The categorization performance in Reuters-21578 dataset when dimension is different. Note that the highest accuracy is highlighted with bold font

dimension	kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
1000	64.446	86.337	68.364	87.100
2000	64.473	84.375	67.773	84.847
3000	63.525	84.012	67.675	84.375
4000	63.977	83.830	65.696	84.230
5000	63.177	83.539	65.755	84.230
6000	63.250	83.503	65.728	84.048
7000	63.412	83.612	65.623	84.121
8000	64.030	83.576	65.853	83.975
9000	63.209	83.321	65.751	83.939
10000	62.829	83.031	65.896	83.830

Note that in Table 3 and Table 4, k takes 45. From Table 5, we can see that when dimension is 1000, the categorization performance of kNN and fkNN reaches the best. This result is very accordant with our dimension design formula:

$$\left\lceil \frac{\lfloor \log(619) \rfloor}{\lfloor \ln(59) \rfloor} \right\rceil \times 1000 = 1 \times 1000 = 1000.$$
 From Table 6, we can find that when dimension is 1000, kNN and fkNN's categorization performance is the best. This result is consistent with our dimension formula design too. We use our dimension formula design in other author's kNN and its variant experiment [13][14][21]. It has the same good result of fitting our dimension formula. Even though it can not get the best point, it must be the better point. The experiment result shows that this dimension formula design is effective and feasible.

6 Conclusion

In this paper, we mainly discuss the improvement of decision rule and design a new algorithm of fkNN (fuzz kNN) to improve categorization performance when the class distribution is uneven. Based on this, we study the selection of dimensions and design a dimension selection formula. The experiment proves that our method is effective.

In the future, we need to study further on how to select the k ; the impact of value k to dimension selection; how to improve the decision rule further, what their effects to be on each other and so on.

Acknowledgment

This research is partly supported by Beijing Jiaotong University Science Foundation under the grant 2004RC008.

References

1. Cover, T. M., Hart P. E.: Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*. Vol. IT-13 (1967) 21-27, 1967
2. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*. Vol. 1 (1997) 76-88
3. Yang, Y., Lin X.: A Re-examination of Text Categorization Methods. In: *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in the Information Retrieval*. ACM Press, New York (1999) 42-49
4. Masand, B., Lino, G., Waltz, D.: Classifying news stories using memory based reasoning. In: *15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval*. Copenhagen (1992) 59-64
5. Lewis, D. D.: Naïve (Bayes) at forty: the independence assumption in information retrieval. In: *Proc. of the 10th European Conference on Machine Learning*. Chemnitz (1998) 4-15
6. Mccallum, A., Nigam K.: A comparison of event models for naïve bayes text classification. In: *AAAI-98 Workshop on Learning for Text Categorization*. Madison, Wisconsin (1998) 41-48
7. Lewis, D. D., Ringuette M.: Comparison of two learning algorithms for text categorization. In: *Proc. of the Third Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas (1994) 81-93
8. Apte, C., Damerau, F., Weiss, S.: Text mining with decision rules and decision trees. In: *Proc. of the Conference on Automated Learning and Discovery, Workshop 6: Learning from Text and the Web*. CMU (1998) 487-499
9. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: *Proc. of the 10th European Conference on Machine Learning*. Chemnitz (1998) 137-142
10. Yang, Y., Chute, C. G.: An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information System*. Vol. 12 (1994) 252-277
11. Ng, H. T., Goh, W. B., Low, K. L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: *20th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval*. (1997) 67-73

12. Wiener, E., Pedersen, J. O., Weigend, A. S.: A neural network approach to topic spotting. In: Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval. (1995) 317-332
13. Tan, S.: Neighbor-weighted K-nearest neighbor for unbalanced text corpus. Expert Systems with Application. Vol. 28 (2005) 667-671
14. Han, E., Karypis, G., Kumar, V.: Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification. In: Proc. of 5th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. (2001) 53-66
15. Shankar, S., Karpis, G.: A Feature Weight Adjustment Algorithm for Document Categorization. In: Proc. of the International Workshop on Multimedia Data Mining. (2000)
16. Li, B., Lu, Q., Yu, S.: An Adaptive k-Nearest Neighbor Text Categorization Strategy. ACM Transactions on Asian Language Information Processing. Vol. 3 (2004) 215-226
17. Lim, H.: An Improved KNN Learning Based Korean Text Classifier with Heuristic Information. In: Proc. of the 9th International Conference on Neural Information Processing. (2002) 731-735
18. Dubois, D., Prade, H.: Fuzzy sets and systems (Theory and application). Oxford, Uk, Academic Press. (1980)
19. Zhao, S.: The method of fuzzy mathematics in pattern recognition. School of the West-North Electronic Engineering Press, Xi'an. (1987)
20. Bian, J., Zhang, X.: Pattern recognition. Tsinghua University Press, Beijing (2000)
21. Cardoso-Cachopo, A., Oliveira, A. L.: An empirical comparison of text categorization methods. In Proceedings of the 10th International Symposium on String Processing and Information Retrieval (SPIRE'03), number 2857 in Lecture Notes in Computer Science, Springer Verlag. (2003) 183-196

Agent-Based Approach for Distributed Intrusion Detection System Design*

Krzysztof Juszczyszyn, Ngoc Thanh Nguyen, Grzegorz Kolaczek, Adam Grzech, Agnieszka Pieczynska, and Radosław Katarzyniak

Institute of Information Science and Engineering, Wrocław University of Technology,
Wyb. Wyspińskiego 27, 50-370 Wrocław, Poland
krzysztof.juszczyszyn@pwr.wroc.pl

Abstract. The aim of this paper is to propose an architecture of distributed Intrusion Detection System (IDS). It is assumed that IDS system will detect and track dissemination and activity of the Internet worms. A general architecture for such a distributed multiagent system is proposed and the tasks, techniques and algorithms to be used are sketched.

1 Introduction

The aim of Intrusion Detection Systems (IDS) is to recognize and notify administrator of the system of various security events as well of incidents and anomalies that can be observed in the context of user behavior or system's and its environment states. All circumstances that indicate security policy violation must be evaluated by IDS due to its role to prevent and reduce the range of unauthorized usage of the system resources. Malicious traffic in the form of Internet worms, denial of service attacks, ports scans etc. became one of the most crucial problems for all that use Internet for communication and so it is also one of the most important challenges for intrusion detection systems. That's why a search for efficient mechanisms to protect networks from malicious traffic is in the main stream of interest of many research and operational communities. Only distributed monitoring and processing of a set of values related to some fundamental network traffic and communication invariants may create an appropriate starting point to evaluation of network security level.

Due to size and dynamics of the modern wide area network systems a distributed multiagent architecture seems to be an effective solution [6]. The system proposed in this paper falls into that class and is intended to allow fast detection and analysis of the ongoing attacks. It is assumed that security system must be able to recognise abnormal states of the network system and point out the sources of worm attack, which is not a trivial task in distributed network environment [10].

In the paper a general architecture of the multiagent system is proposed (Sec. 2), next the tasks assigned to the classes of agents in the system are defined (Sec. 3). The information integration processes are defined both on the level of monitoring agents (Sec. 4) and managing agents (Sec. 5) which infer about global anomalies and their sources.

* This work was supported by the Polish State Committee for Scientific Research under Grant No. 3 T11C 029 29 (2005-2007).

2 The Architecture of the Multi-agent System

The aim of this paper is to propose a framework for distributed multiagent Intrusion Detection System (IDS). It is assumed that the network system is consisted of the set of nodes. There are also two types of agents in our multiagent system: monitoring agents (MoA) and managing agent (MA). Monitoring agents (MoA) observe the nodes, process captured information and draw conclusions that are necessary to evaluate the current state of system security (Fig. 1). Each agent MoA monitors its own area of responsibility consisting of the set of nodes. It is assumed that these areas may mutually overlap [6].

It is known that in the case of worm attack there occur at least two kinds of anomalies: in observed traffic characteristics and in communication scheme which tends to be constant under normal conditions (see sec. 4.2). The system properties observed by the agent MoA fall into two basic (and physically different) categories: 1. Traffic measurement. 2. Communication pattern measurement.

In order to store and process information about the nodes under control each MoA agent uses two matrices: Traffic Matrix M_t and Communication Matrix M_c . Both have $n \times n$ size where n is the number of nodes controlled by a MoA agent. The rules for setting their values are given in the next sections.

Our architecture is intended to allow worm propagation detection and worm-based attacks on the basis of the fusion of information gathered from matrices M_t and M_c of all MoA agents.

In our approach following techniques are used: 1. Standard traffic analysis methods for traffic measurement. 2. Graph theoretic methods to identify abnormal communication patterns. 3. Consensus and data fusion methods to manage the system and generate alerts. The basic functional structure of monitoring and managing agents is shown on the Fig.2.

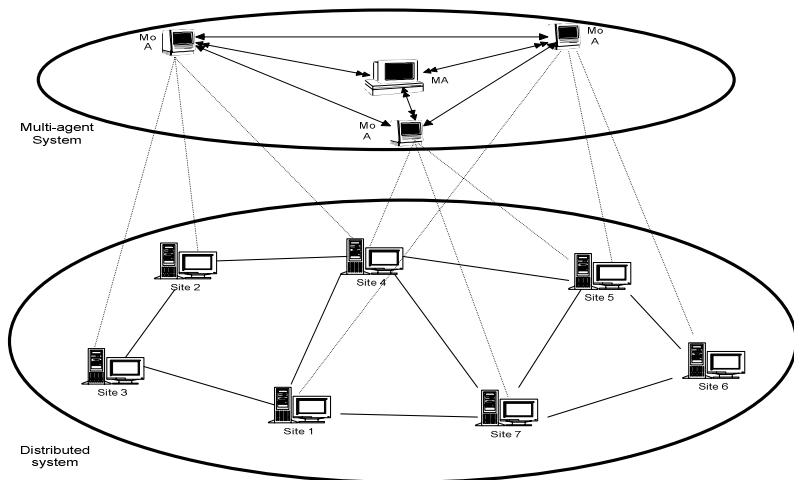


Fig. 1. The conception of the multi-agent system

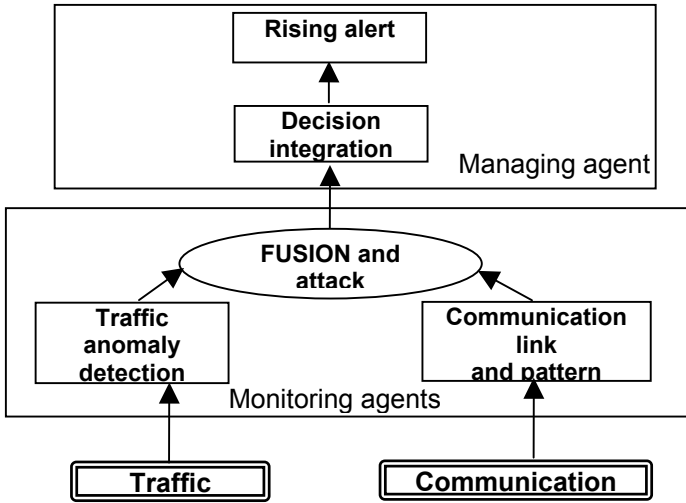


Fig. 2. The functional structure of monitoring and managing agents

3 The Main Tasks of the Intrusion Detection System

The multi-agent intrusion detection system has a number of tasks. Monitoring agents have similar tasks but managing agent(s) has different tasks. The tasks referring to work organization, knowledge processing, ontology integration have been outlined in work [6]. In this paper we present the description of the main task of the system, which is relied on detecting the resource of the attack as well as the kind of the attack. Denote by A the set of agents in the multi-agent systems; S – the set of sites of monitored system and T – the set of discrete ordered moments of time. To realize this task the system uses monitoring agents and one or more managing agents. This task is divided into following subtasks:

1. As mentioned above, a monitoring agent A_i observes the nodes in his region and analyzes gathered information in order to determine a tree $T_i^{(j)} = (S_i, R_i^{(j)})$, where $S_i \subset S$, $R_i^{(j)}$ is a binary relation in S_i such that pair $\langle s, s' \rangle \in R_i^{(j)}$ for $s, s' \in S_i$ if and only if in the opinion of agent A_i the attack has come directly from site s to s' .
2. The decisions of monitoring agents are successively sent to the managing agent. The task of this agent is to integrate these decisions in order to determine the global tree representing the propagation situation of the attack in the whole system. Owing to this global tree one should get to know the resource of the attack as well as its propagation plan. For realizing this task the managing agent should have the following tools:
 - Criterion for assessment if the set of trees representing the knowledge states of monitoring agents in a time interval is enough for making a sensible decision.
 - An effective algorithm for integrating given trees and determining a global tree.

For the first task a conception of an algorithm is presented in Section 4.3. Section 5 contains some idea for the second task. These tasks have been discussed in detail in numerous meetings by our project team consisting of the authors of work [6].

4 Making Decision Process of Monitoring Agents

4.1 Traffic Analysis

Network traffic anomalies are one of the most important sources of information for contemporary Intrusion Detection Systems. Traffic anomalies typically are related to states when network processes deviate from normal behavior. These anomalous events will disrupt the normal behavior of some measurable network data.

Anomaly detection system requires precise characteristic of network behavior and specially definition of parameters values related to the normal traffic. Selection of substantial set of variables for traffic description depends on several network specific factors such as the dynamics of the network being studied in terms of traffic volume, the type of network data available, and types of applications running on the network.

There are a few main streams that can be observed in the literature as the online modeling of network traffic, parsimonious traffic models that accurately capture fractal and multi-fractal scaling properties, such as the self-similar models introduced by Norros [9]. Lakhina et al. also have shown the application of Principal Component Analysis to separate of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions [10].

In our approach, we only consider one source type of network traffic anomalies - malicious traffic. We assume that the Intrusion Detection System should recognize anomalies related to phenomenon like self-propagating worms, viruses, port scans and denial of service attacks. The choosing of certain traffic analysis method is now an open question and depends on the first assumptions concerning simulation and testing framework. However, a MoA will track traffic anomalies within its area of responsibility and the results of the tracking will be stored in a traffic matrix M_t , which will have $n \times n$ size where n is a number of network nodes belonging to MoA's area of responsibility.

The values of M_t matrix are to be set according to the following rules:

$$M_t(i, j) = \begin{cases} 0 & : \text{there is no data link between nodes } i \text{ and } j \\ 1 & : \text{nodes } i \text{ and } j \text{ communicate} \\ \varepsilon & : \text{link state is unknown} \\ a_k & : \text{anomaly of type } k \text{ was detected between } i \text{ and } j \end{cases}$$

Where a_k (anomaly type) is intended to identify traffic anomaly by means of the method of detection or magnitude. This information maybe important for monitoring agent's decision making process. The M_t matrix of any given MoA will be periodically sent to managing agent AM in order to reason about global traffic disturbances caused by the worm attack.

4.2 Communication Patterns

4.2.1 Communication Patterns – State of the Art

Recent results show that network traffic show some quantitative and topological features that appear to be invariant and characteristic for given network. Moreover, general rules underlying that features are the same for almost any network of remarkable size. These distinct features concern topology of network communication, considered as origin-destination flows graph, the distribution of data volumes sent between destinations and the in/out ratio of data sent between subnets and outside world.

The general idea of detecting worm attack by observation of communication patterns is to compare patterns existing under normal state of the network to new ones which occur under attack. In both cases there exist several data flows in the network (origin-destination flows). Within a network under attack there are also scanning and attack flows which differ substantially from normal network activity [8]. Moreover, total scanning rate into the sub-network (or given set of nodes) is a function of the number of all infected nodes in the network.

Also, proportion between a number of internal and external data flows for a given subnet is constant for a long time periods and different scales (subnet sizes) or traffic types (protocols) [1]. For any network node its fan-in (percent of hosts that originate conversations with that node) and fan-out coefficients are constant both for local and wide-area traffic.

For communication patterns in entire network there are also some interesting results. In particular, it was proved that the IP graph has heavy-tailed degree distribution showing scale-free structure according to power law (probability $P(k)$ of the node having degree of k equals approximately $k^{-\gamma}$) [4]. Under worm attack the structure of communication is heavily affected and the power law distribution changes. There is also a detectible dependence between worm propagation algorithm, scale of the attack and communication pattern disturbance [5].

Similar relationships occur also on the level of given communication protocol, for example after investigation of the topology of e-mail corporate network (with e-mail addresses as nodes and e-mails as links) it was found that the resulting network exhibits a scale-free link distribution and small-world behaviour, as for known social networks [3]. This result was then used to propose an anti-spam tool [2].

4.2.2 Detectable Communication Patterns

Monitoring agents of proposed IDS system will gather information about communication within the network under control, then the existing communication patterns will be discovered (global patterns are by definition visible only for monitoring agents). The system will be viewed as a graph consisting of nodes (each monitoring agent will have a set of nodes under control) and edges which appear if there exists data flow between given pair of nodes. In our approach we postulate tracking the following communication patterns:

- Node degree distribution (global pattern)
- Clustering coefficient for a given node (local, detected on global level)
- Fan-in and Fan-out ratios (local, detected on global level)

All the above patterns are (according to results listed in previous section) invariant during most time of normal system activity. But under worm attack they will change leading to alert and taking chosen countermeasures.

Each MoA agent stores data about communication in the form of M_c matrix. The values of M_c are set according to the following rules:

$$M_c(i, j) = \begin{cases} 0 & : \text{there is no communication between nodes } i \text{ and } j \\ 1 & : \text{nodes } i \text{ and } j \text{ communicate} \\ \varepsilon & : \text{no knowledge about communication between } i \text{ and } j \end{cases}$$

1. *Node degree distribution* is computed after gathering information in form of M_t matrices from MoA agents.

2. The *clustering coefficient* C is the probability that two nearest neighbours of vertex i are also neighbours of each other. C provides a quantitative measure for cliques in communication graph. For node i clustering C_i is given by:

$$C_i = \frac{2k_i}{n_i(n_i - 1)}$$

where n_i is the number of its neighbours and k_i – the number of connections between them. Thus high C reflects that a node belongs to a clique. There is also a power-law distribution of C values reflecting the fact that nodes with high degree (hubs) show low values of clustering coefficient (they connect cliques).

3. Fan-in and Fan-out ratios are computed by the managing agent:

Fan-in is the number of nodes that originate conversations with node i , while Fan-out is the number of hosts to which i initiates conversations. Experiments showed that both Fan-in and Fan-out for given node and their distribution for all nodes tend to be constant under normal conditions.

4.3 The Outline of Algorithm for Monitoring Agent Decision Making Process

The MoA agent's algorithm for decision making process is invoked periodically and uses values of M_t and M_c matrices as input data. It should be noted the MoA stores acquired values of M_t and M_c thus creating the history of system behaviour. The algorithm itself consists of the following steps:

Given: Matrices M_t and M_c (as defined in Sec. 4.1 and 4.2.2).

Result: The tree $T_i^{(j)} = (S_i, R_i^{(j)})$ (see Sec. 3).

BEGIN

1. Detect traffic anomalies (using chosen technique). Fill cells in M_t .
2. Create a list of traffic anomalies (on the basis of the values from M_t).
3. Compute the communication patterns (on the basis of the values from M_c).
4. Create a list of communication anomalies (where a communication anomaly is unexpected or rapid change of clustering, fan-in/fan-out ratio for a given node or a change in a node degree distribution).
5. If any of the anomalies' lists is not empty, perform an attack backtracking analysis which will return result in the form of attack tree $T_i^{(j)} = (S_i, R_i^{(j)})$ as defined in Sec. 3. Note, that constructing such a graph requires data fusion from

two different sources – traffic and communication anomalies detection. The joint usage of two physically different groups of measures will provide more accuracy in tracking attacks, which may be guided by different worm propagation algorithms.

END.

Note also that some fields in each M_t and M_c may be unknown (if their actual values were for some reason not observed by the MoA) which result in some uncertainty in attack investigation analysis. The level of this uncertainty will be also a part of the algorithm's result.

5 Integration of Monitoring Agents' Decisions

In this section we present an outline of the realization scheme for the second task defined for the managing agent. As mentioned, the managing agent successively obtains from monitoring agents their decisions in form of trees, which are related to particular moments of time. It should have a criterion for assessing the susceptibility of these trees to a credible global tree and an effective algorithm for its determining.

Formally, we may assume that for given time interval $[t, t']$ in the database of the managing agent there is a set of trees, that comes from one monitoring agents:

$$T_{[t, t']} = \{T_i^{(j)} = (S_i, R_i^{(j)}): i=1, \dots, m \text{ and } t \leq t_j \leq t'\}.$$

It seems to be intuitive that the special subject of interest of the agent is such set $T_{[t, t']}$, where t' is the current time moment. For this set the mentioned criterion for credibility susceptibility (CS) can be formulated in the following way:

Let $G_i^{(j)} = (S_i, V_i^{(j)})$ be a non-directed graph which arises from tree $T_i^{(j)}$ in such way that it contains the same edges as tree $T_i^{(j)}$, but these edges are non-directed. Let

$$S = \bigcup_{i=1}^n S_i \text{ and } V = \bigcup_{i=1}^n V_i.$$

The criterion CS can be defined as: *Non-directed graph (S, V) should be coherent.*

The intuition of this condition is that the knowledge of monitoring agents referring to attacked sites is in some sense complete. It means that if (in opinions of some monitoring agents) two sites are attacked then there is a relationship between them. This condition is good only if only one site is the source of investigated attack. We accept this assumption because it is the most common case in practice.

If the set $T_{[t, t']}$ of trees satisfies the condition included in criterion CS then the managing agent MA can use an algorithm for determining the global tree representing the attack propagation. The idea of this algorithm is based on consensus methods [7]. The sketch of this algorithm is as follows:

Given: Set $T_{[t, t']}$ of trees satisfying the condition included in criterion CS.

Results: Tree (S, V) representing the global attack propagation

BEGIN

1. Let $S = \bigcup_{i=1}^n S_i$ and $V = \emptyset$;
2. Complete V using postulates *Closure of knowledge*; *Consistency of knowledge* and *Superiority of knowledge* defined in work [7].
3. For such edges referring to which some of monitoring agents are in conflict, use a consensus method for solving.
4. If there is a lack of edges for creating a tree, ask monitoring agents MoA for completing data.

END.

6 Conclusions

In this paper a distributed, multi-agent IDS design conception has been presented. This conception takes into account many symptoms of the ongoing worm attacks. The same architecture may be as well adapted to the control network communication limited to chosen protocol or service (described regularities might be applied on different levels: IP traffic, protocols, etc.). The future works should concern the detailed design of the system and working out algorithms for its functioning.

References

1. Allmanz M. et.al. A First Look at Modern Enterprise Traffic, In Proc. Internet Measurement Conference, October 2005, 217-231.
2. Boykin O., Roychowdhury V. Personal Email Networks: An Effective Anti-Spam Tool, IEEE Computer, Vol. 38, No. 4, (2005), 61-68.
3. Ebel H., Mielsh L., Bornholdt S., Scale-free topology of e-mail networks, Physical Review E 66, 2002, 121-131.
4. Faloutsos M., Faloutsos P., Faloutsos C., On power-law relationships of the Internet topology. In Proc.ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 1999, 251-262.
5. Kohler E., Liy J., Paxson V., Shenker S., Observed Structure of Addresses in IP Traffic, In Proc. SIGCOMM Internet Measurement Workshop, November 2002, 253 - 266.
6. Kolaczek G., Kuchtiak-Pieczynska A., Juszczyzyn K., Grzech A., Katarzynak R., Nguyen N.T. (2005): A Mobile Agent Approach to Intrusion Detection in Network Systems. In: Proceedings of KES 2005, Lecture Notes in Artificial Intelligence 3682 (2005) 514-519.
7. Nguyen N.T.: Consensus systems for conflict solving in distributed systems. Information Sciences 147 (1-4) (2002) 91-122
8. Nicol D., Liljenstam M., Liu J., Multiscale Modeling and Simulation of Worm Effects on the Internet Routing Infrastructure, In Proc. Performance Tools Conference, 2003, 1-10.
9. Norros I., A storage model with self-similar input, Queueing Syst. 16 (1994) 387-396,.
10. Lakhina A., Crovella M., Diot. C. Diagnosing Network-Wide Traffic Anomalies. In Proc. of ACM SIGCOMM 2004. Portland. (2004) 219 - 230.

A Novel Approach for Similarity Measure Schemes Based on Multiple Moving Objects in Video Databases

Choon-Bo Shim¹, Chang-Sun Shin¹, DongGook Park¹, and Won-Ho So²

¹ School of Information & Communication Engineering

² Dept. of Computer Education, Suncheon National University, Suncheon, Jeonnam
540-742, South Korea

{cbsim, csshin, dgpark, whso}@suncheon.ac.kr

Abstract. The general aim of this paper is to study the spatio-temporal modeling techniques which can efficiently represent multiple moving objects' in video databases. The traditional schemes only consider direction property, time interval property, and spatial relationship property for modeling moving objects' trajectories. But, our scheme also takes into account on distance property, conceptual location information, and related object information so that we may improve a retrieval accuracy to measure a similarity between two moving objects as well as them. As its application, we implement the Content- and Semantic-based Soccer Video Retrieval (CS²VR) system by using MS Visual C++ and DirectX for indexing and searching on soccer video data. The CS²VR helps users to easily extract the trajectory information of soccer ball from soccer video data semi-automatically as well as to conveniently retrieve the results acquired by sketching query trajectory with mouse button.

1 Introduction

The initial research issues on the content-based video retrieval have highly concentrated on data representation schemes which can efficiently model content itself extracted from video data [1-5]. However, for handling a large amount of multimedia data, it is required to provide schemes with good retrieval performance on a variety of user queries. Thus, we propose a new spatio-temporal modeling technique which can efficiently represent multiple moving objects in video databases. The traditional schemes only consider direction property, time interval property, and spatial relationship property for modeling moving objects' trajectories. However, our scheme also takes into account on distance property, conceptual location information, and related object information (e.g. player name having a soccer ball) so that we may improve retrieval accuracy to measure a similarity between two moving objects as well as them. Therefore, the proposed spatio-temporal scheme can support content-based retrieval using moving objects' trajectories as well as semantics-based retrieval using concepts which are acquired through the conceptual location information of moving objects. As its application, we design and implement the Content- and Semantic-based Soccer Video Retrieval (CS²VR) system. Finally, in our performance study, our scheme yields substantially better retrieval performance compared to existing related work in term of retrieval effectiveness.

This paper is organized as follows: Section 2 presents a new spatio-temporal modeling scheme for representing multiple moving objects; Based on our modeling scheme, the similarity measure functions for multiple moving objects is described in Section 3; For its applications, the Content-based Soccer Video Retrieval (CS²VR) system is presented in Section 4; and Section 5 covers conclusion with brief summary.

2 Spatio-temporal Modeling Scheme

Since video data have both spatial and temporal information, they should consider both spatial [6] and temporal [7] relationships to represent moving objects in an effective way. For this, we propose a new spatio-temporal representation scheme for modeling the multiple trajectories among several moving objects. At first, spatio-temporal representation scheme for single trajectory composed of only one moving object is introduced.

Definition 1. Motion property information for a moving object A over all the time intervals, MPS(A), is defined as follows:

$$MPS(A) = \{M_i(A) \mid i = 0, \dots, n-1\} = \{M_0(A), M_1(A), \dots, M_{n-1}(A)\}$$

Definition 2. A motion property for a moving object A over time interval I_i, M_i(A), is defined as follows:

$$M_i(A) = (R_i(A), D_i(A), I_i(A))$$

Here, R_i(A) is a moving direction over time interval I_i (= [t_i, t_{i+1}]) and is represented as a real angle with a range of 0 to 360 degree. D_i(A) is a moving distance over I_i and is described as an absolute Euclidean distance or a relative distance. I_i(A) means a time interval from the start time to the end time while the moving object A is moving.

Definition 3. Stationary property information for a moving object A over all the time instances, SPS(A), is defined as follows:

$$SPS(A) = \{S_i(A) \mid i = 0, \dots, n\} = \{S_0(A), S_1(A), \dots, S_n(A)\}$$

Definition 4. A stationary property for a moving object A at time t_i, S_i(A), is defined as follows:

$$S_i(A) = ([L_i(A)], [O_i(A)])$$

Here, L_i(A) is a location information of the moving object A. The location information describes a real location in coordinates or a semantic-based location according to a real application, e.g., penalty area or goal area in the soccer game. O_i(A) is an object information related with the moving object A, e.g., actor or owner having the moving object A. Here, [] means an optional operator.

For the single trajectory of a moving object A, it is possible to combine a motion property (Definition 1 and 2) with a stationary property (Definition 3 and 4). As a result, the information of a single trajectory is defined as follows.

Definition 5. For a given ordered list of time interval I_0, I_1, \dots, I_{n-1} , the single trajectory information of a moving object A, $ST(A)$, is defined as follows:

$$ST(A) = MPS(A) + SPS(A)$$

We define multiple trajectories as the trajectories of two or more moving objects. However, since the multiple trajectories can be represented by the combination of the trajectory between two moving objects, we first define a relationship trajectory between two objects.

Definition 6. Let at least one of object A and object B be a moving object. Motion property information for A and B over all the time interval, $MPM(A, B)$, is defined as follows:

$$MPM(A, B) = \{M_i(A, B) \mid i = 0, \dots, n-1\} = \{M_0(A, B), M_1(A, B), \dots, M_{n-1}(A, B)\}$$

Definition 7. Let at least one of object A and object B be a moving object. A motion property for A and B over time interval $I_i ([t_i, t_{i+1}])$, $M_i(A, B)$, is defined as follows:

$$M_i(A, B) = (D_i(A, B), I_i(A, B))$$

Here, $D_i(A, B)$ is a relative moving distance of A to B over I_i and is ranged from 0 to 100. That is, $D_i(A, B)$ is 50 in case the moving distance of A is the same as that of B. $D_i(A, B)$ is ranged from 51 to 100 in case the moving distance of A is greater than that of B while it is near to 0 as the moving distance of A is less than that of B. $I_i(A, B)$ is the same as single trajectory.

Definition 8. Let at least one of object A and object B be a moving object. Stationary property information for A and B over all the time instances, $SPM(A, B)$, is defined as follows:

$$SPM(A, B) = \{S_i(A, B) \mid i = 0, \dots, n\} = \{S_0(A, B), S_1(A, B), \dots, S_n(A, B)\}$$

Definition 9. Let at least one of object A and object B be a moving object. A stationary property for A and B at time t_i , $S_i(A, B)$, is defined as follows:

$$S_i(A, B) = ([L_i(A)], [O_i(A)], ([L_i(B)], [O_i(B)], T_i(A, B), R_i(A, B))$$

Here, $L_i(A)$ and $L_i(B)$ are the location information of moving object A and B, respectively. $O_i(A)$ and $O_i(B)$ are the actors having moving objects A and B, respectively. $T_i(A, B)$ is a spatial (topological) relations on XY-coordinates from A to B, being represented as one of seven topological relations operator : FA(FarAway), DJ(DisJoint), ME(MEet), OL(OverLap), CL(is-inCLuded-by), IN(INclude), and SA(SAme). Finally, $R_i(A, B)$ means a directional relations from A to B and is ranged from 0 to 360 degree.

For a relationship trajectory between A and B, it is possible to combine a motion property (Definition 6 and 7) with a stationary property (Definition 8 and 9). As a result, the relationship trajectory information is defined as follows.

Definition 10. Let at least one of object A and object B be a moving object. For a given ordered list of time interval I_0, I_1, \dots, I_{n-1} , the relationship trajectory information between A and B, $RT(A, B)$, is defined as follows:

$$RT(A, B) = MPM(A, B) + SPM(A, B)$$

Based on Definition 5 and 10, the multiple trajectory information of two or more moving objects, $MT(A_1, A_2, \dots, A_n)$, can be represented by a combination of the relationship trajectory information (RT) and the single trajectory information (ST).

Definition 11. Among objects A_1, A_2, \dots, A_n , let i be the number of moving objects and j be the number of stationary objects, i.e., $n=i+j$. The multiple trajectory information of A_1, A_2, \dots, A_n , $MT(A_1, A_2, \dots, A_n)$, is defined as follows:

$$MT(A_1, A_2, \dots, A_n) = \{ST(A_p) \mid p = 1, \dots, i\} + \{RT(A_q, A_{q+1}) \mid q = 1, \dots, k\} \quad , k = {}_n C_2 - j C_2$$

Here $ST(A_i)$ is the single trajectory information of an object A_i . $RT(A_k, A_{k+1})$ is the relationship trajectory information between object A_k and A_{k+1} where k is the number of relationship trajectories between two moving objects as well as between a moving object and a stationary object.

3 Similarity Measure

Based on our spatio-temporal modeling for moving objects’ trajectories, we define a similarity measure for a single trajectory and multiple trajectories, respectively. Since we measure a similarity between i -th motion in query trajectory Q and j -th motion in data trajectory S , we define a distance function between two motions.

Definition 12. A distance function, $dS_{df}(q[i], s[j])$, to measure the similarity between the arbitrary motion $s[i]$ of a data trajectory S and the arbitrary motion $q[j]$ of a query trajectory Q is defined as follows.

$$d_{dis}(s[i,2], q[j,2]) = |s[i, 2] - q[j, 2]|$$

$$\text{if } |s[i, 1] - q[j, 1]| > 180 \text{ then } d_{ang}(s[i, 1], q[j, 1]) = (360 - |s[i, 1] - q[j, 1]|)$$

$$\text{else } d_{ang}(s[i, 1], q[j, 1]) = |s[i, 1] - q[j, 1]|$$

$$dS_{df}(s[i], q[j]) = ((d_{ang} / 180) * \alpha) + ((d_{dis}/100) * \beta)$$

Here, d_{ang} is a distance function for the direction (angle) property for all the motions of a trajectory and d_{dis} is a distance function for the distance property. $s[i, 1]$ and $s[i, 2]$ are the direction and the distance value of the i -th motion in a trajectory S , respectively. α and β mean the weight of the direction and the distance, respectively, when $\alpha+\beta=1.0$.

Definition 13. Given a single trajectory $S=\{s[0], s[1], \dots, s[m]\}$ and a query trajectory $Q=\{q[0], q[1], \dots, q[n]\}$ ($1 \leq n \leq m$), the similarity between trajectory S and Q , $ST_{sim}(S, Q)$, by using definition 12 is defined as follows.

$$ST_{sim}(S, Q) = \text{MAX}_{i=1}^n \left\{ \frac{\sum_{j=1}^n dS_{df}(s[i + j], q[i])}{n} \right\} \quad (\forall_j, 0 \leq j \leq m - n)$$

Next, to measure similarity between multiple trajectories, we first define the similarity measure for a relationship trajectory between two trajectories. For this, we make use of topological relationships between multiple moving objects as well as moving direction and moving distance. Hence, we define a distance function using three-dimensional properties as follows.

Definition 14. A distance function, $dR_{df}(q[i], s[j])$, to measure the similarity between the arbitrary motion $s[i]$ of a data trajectory S and the arbitrary motion $q[j]$ of a query trajectory Q is defined as follows.

$$\begin{aligned}
 d_{top}(s[i,3], q[j,3]) &= (\text{top_dist}(s[i, 3], q[j, 3]))^2 \\
 d_{dis}(s[i,2], q[j,2]) &= |s[i, 2] - q[j, 2]| \\
 \text{if } |s[i, 1] - q[j, 1]| > 180 &\text{ then } d_{ang}(s[i, 1], q[j, 1]) = (360 - |s[i, 1] - q[j, 1]|) \\
 &\text{else } d_{ang}(s[i, 1], q[j, 1]) = |s[i, 1] - q[j, 1]|
 \end{aligned}$$

$$dR_{df}(s[i], q[j]) = ((d_{ang} / 180) * \alpha) + ((d_{dis}/100) * \beta) + ((d_{top}/25)*\gamma)$$

Here d_{ang} is a distance function for the direction (angle) property for all the motions of a trajectory, d_{dis} is a distance function for the distance property and d_{top} is a distance function for the topology property. $s[i, 1]$, $s[i, 2]$ and $s[i, 3]$ are the angle, the distance and the topology value of the i -th motion in a multiple trajectory S , respectively. α , β and γ mean the weight of the angle, the distance and the topology, respectively, when $\alpha+\beta+\gamma=1.0$. Also, $\text{top_dist}(a, b)$ means the similarity distance for topological relations between a and b .

Definition 15. Given a relationship trajectory $S=\{s[0], s[1], \dots, s[m]\}$ and a query trajectory $Q=\{q[0], q[1], \dots, q[n]\}$ ($1 \leq n \leq m$), the similarity between trajectory S and Q , $RT_{sim}(S, Q)$, by using definition 14 is defined as follows.

$$RT_{sim}(S, Q) = \text{MAX}_{j=1}^n \left\{ \frac{\sum_{i=1}^n dR_{df}(s[i + j], q[i])}{n} \right\} \quad (\forall j, 0 \leq j \leq m - n)$$

A list of multiple trajectories of at least two or more moving objects, $MT(A_1, A_2, \dots, A_n)$, can be represented as a combination of single trajectory(ST) and relationship trajectories(RT).

Definition 16. Let us suppose that i and j is the number of moving objects and stationary objects, respectively ($n=i+j$). The similarity for multiple trajectories of objects A_1, A_2, \dots, A_n , $MT(A_1, A_2, \dots, A_n)$, is calculated as follows:

$$MT_{sim}(A_1, A_2, \dots, A_n) = \frac{\sum_{p=1}^i ST_{sim}(A_p) + \sum_{q=1}^k RT_{sim}(A_q, A_{q+1})}{i + k} \quad , k = {}_n C_2 - {}_j C_2$$

Here $ST_{sim}(A_i)$ is the similarity value of single trajectory of object A_i . $RT_{sim}(A_k, A_{k+1})$ is the similarity value of relationship trajectories between objects A_k and A_{k+1} where k

is the number of relationship trajectories between two moving objects as well as between a moving object and a stationary object.

4 CS²VR System

We make use of soccer video data as its application since there are many trajectories data of salient objects such as soccer ball, player, and referee in the soccer ground field. Additionally, the trajectory information plays an important role in indexing soccer video data and detecting the scene with ‘goal in’ in soccer video database. Therefore, we implement the Content- and Semantic-based Soccer Video Retrieval (CS²VR) under Windows 2003 Server O.S with Pentium III-800 and 512 MB memory by using Microsoft Visual C++ compiler and DirectX so that we can support the indexing and searching on soccer video data. Figure 1 first shows a Graphic User Interface (GUI) for soccer video indexing which can help users to easily extract the trajectory information of soccer ball from soccer video data semi-automatically, e.g. moving direction, moving distance, # of frame, player name, and so on. We implement it so as to work well under Window platform. Our GUI for soccer video indexing is composed of two windows: main window and soccer ground window. The former is to browse raw soccer video data formatted as mpeg (*.mpeg) and to extract the trajectory information of soccer ball and main player. The latter is to transform the location of soccer ball on raw video through the main window data into an absolute location on the coordinate of soccer ground field. For this, we make use of ‘affine transformation algorithm’ which is mainly used in computer vision or image processing fields [8].

Figure 2 depicts a GUI for soccer video retrieval which can help users to retrieve the results acquired in soccer video databases. We implemented it by JAVA application in order to work well independently without regard to system platform. We can

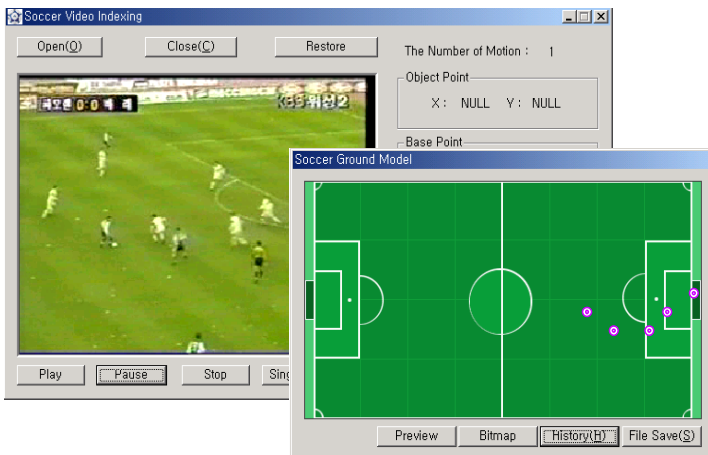


Fig. 1. GUI for soccer video indexing in CS²VR system

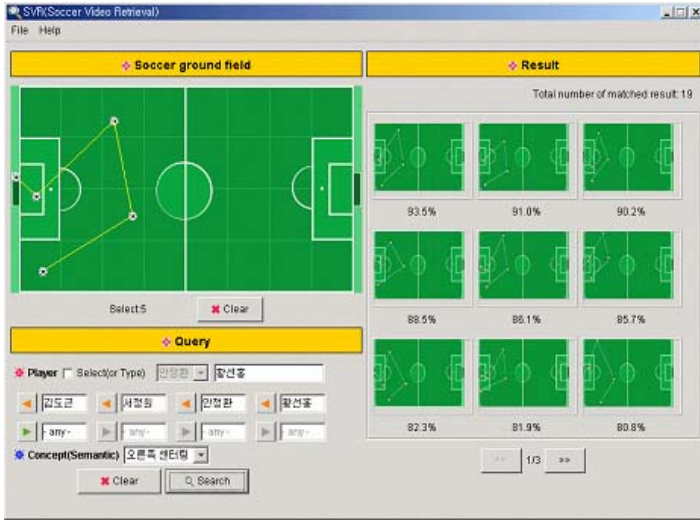


Fig. 2. GUI for soccer video retrieval in CS²VR system

provide three types of user query, that is, trajectory-based query, semantic-based query, and actor-based query as shown in the left part of Figure 2. The trajectory-based query is based on the trajectory of moving objects such as soccer ball and player as the following query: “Finds all video shots whose trajectory is similar to the trajectory sketched by a user on soccer video retrieval interface”. The semantic-based query is based on important semantics such as ‘penalty kick’, ‘corner kick’ and ‘goal in’ in soccer video databases: “Finds all video shots including a scene ‘goal in’”. Finally, the actor-based query is based on the interested player name such as ‘Ronaldo’, ‘Rivaldo’ and ‘Zidane’ in soccer video databases: “Finds all video shots including a scene ‘goal in’ by player name ‘Ronaldo’”. The retrieved results on a user query are provided in the form of trajectory images with the similar trajectory as shown in the right part of Figure 2. We can browse them in the order of the degree of relevance to a user query. The real soccer video shot corresponding to the trajectory image retrieved can be shown by clicking its trajectory image with the mouse button.

5 Conclusions

We proposed a new spatio-temporal modeling scheme which can efficiently represent multiple moving objects in video databases. In order to improve retrieval accuracy to measure a similarity between two moving objects, our scheme took into account on distance property, conceptual location information, and related object information while the traditional schemes only consider direction property, time interval property, and spatial relations property for modeling moving objects' trajectories. We designed and implemented the Content- and Semantic-based Soccer Video Retrieval (CS²VR) system as its application. As future study, it is necessary to research on indexing technique to support good retrieval efficiently when the amount of multiple moving

objects is very large and STQL(Spatio-Temporal Query Language) to make query formally for moving objects in spatio-temporal databases like SQL.

Acknowledgements

This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Industrial-Academic Cooperation Centered University.

References

- [1] J. R. Smith, S. F. Chang, "VisualSEEK: a Fully Automated Content-Based Image Query System," in Proceedings of ACM Multimedia 96, pp. 87-98, 1996.
- [2] Virginia, E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of images," IEEE Computer, Vol. 28, No. 9, pp. 40-48, 1995.
- [3] G. Ahanger, D. Benson, and T. D. C Little, "Video query formulation," in Proceedings of SPIE Electronic Imaging Science and Technology, pp. 280-291, 1995.
- [4] A. Yoshitaka, M. Yoshimitsu, M. Hirakawa, and T. Ichikawa, "V-QBE: Video database retrieval by means of example motion of objects," in Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp. 453-457, 1996.
- [5] Z. Aghbari, K. Kaneko, and A. Makinouchi, "Modeling and Querying Videos by Content Trajectories", In Proceedings of the International Conference and Multimedia Expo, pp. 463-466, 2000.
- [6] J. W. Chang, Y. J. Kim, and K. J. Chang, "A Spatial Match Representation Scheme Indexing and Querying in Icnioic Image Databases," ACM International Conference on Information and Knowledge Management, pp. 169-176, 1997.
- [7] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," Communication of the ACM, Vol. 26, No. 11, pp. 832-843, 1983.
- [8] H.S. Yoon, J. Soh, B.W. Min, and Y.K. Yang, "Soccer image sequences mosaicing using reverse affine transform," In Proc. of International Technical Conference on Circuits/Systems, Computers and Communications, pp. 877-880, 2000.

An Ontology for Network Services

Pedro Alípio, José Neves, and Paulo Carvalho

Universidade do Minho, Departamento de Informática,
4710-057 Braga, Portugal
{pma, jneves, pmc}@di.uminho.pt

Abstract. Most of the network service specifications use XML based models. However, as XML imposes a hierarchical structure, several types of relations may not be modeled. Therefore, richer specification languages are required in order to specify all network services vocabulary and how it relates with management tasks and with network configuration. This paper presents an ontology based model for network services, overcoming those semantic gaps and creating a better ground for reasoning over services fostering their self-configuration.

1 Introduction

Many network service management tasks such as service administration, service quality monitoring, service configuration, and resource optimization are often performed manually. This work can be time-consuming and very sensible to human errors. Moreover, it requires a growing number of highly skilled personnel, bringing huge costs to Internet Service Providers (ISPs).

Frequently, ISPs network services are expressed through Service Level Agreements (SLAs), where a technical part called Service Level Specification (SLS) is included. Several proposals of SLA and SLS specification have been presented, fostering a common ground for interoperability among domain and interdomain network service configuration agents. However, none of those specification is expressive enough to include the necessary knowledge to map service requirements into network configurations.

An ontology defines a common vocabulary for information interchange in a knowledge domain [1] and allows: (i) sharing common understanding of the structure of information among people or software agents; (ii) the reuse of domain knowledge; (iii) making domain assumptions explicit; (iv) separating domain knowledge from the operational knowledge; (iv) analysing domain knowledge [2]. This work suggests the use of an ontology for ISP network service specification instead of the traditional approaches.

This paper has the following structure: related work and the state-of-the-art in network service specification is presented in Section 2; the concepts and relations used to model service vocabulary and configuration mappings are explained in Section 3; finally, conclusions and future work are presented in Section 4.

2 Related Work

Network service specifications, apart from being a key aspect for QoS provisioning, provide a valuable input for network configuration. Therefore, defining a network service ontology, including SLA semantics and vocabulary is crucial for ensuring Quality of Service (QoS). Several working groups are committed to SLS definition and management [3, 4, 5, 6]. Usually, XML is the preferred network services specification language. However, pure XML forces a hierarchical structure and it does not allow complex relations between objects.

Lately, ontologies are being used to bring semantics to the World-Wide Web (WWW). The WWW Consortium (W3C) developed the Resource Description Framework (RDF) [7], a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. More recently, several ontology specification languages were developed with more expressive constructs aimed at facilitating agent interaction on the Web [8, 9].

Most of the ontology specification languages rely on XML and RDF only as underneath platform [8, 10, 9]. As a result, these ontologies may be validated, parsed or transformed with regular XML tools. Nevertheless, reasoning (queries, verification and taxonomical inference) is often performed by knowledge based systems that use other formalisms.

3 Network Service Specification Ontology

The main objective of the ontological representation of network services is to create a common vocabulary, including a service classification, and to map service attributes into network configurations. The model comprises three abstraction layers: (i) a higher level including the SLA and the service classification; (ii) a medium level including service level specification sections, as it is presented in [6]; (iii) a lower level, including the mappings to network configurations following the guidelines for Differentiated Services (Diffserv) [11].

3.1 Service Classification

Network traffic is classified in three groups: (i) Network Control for routing and network control function; (ii) Operations, Administration and Management (OAM) for network configuration and management functions; and (iii) the User/Subscriber traffic group for ISP functions which may be divided into nine different categories, namely: Telephony service; Signalling service; Multimedia Conferencing service; Real-time Interactive service; Multimedia Streaming service; Broadcast Video service; Low Latency Data service; High Throughput Data service; and Default service class.

Although User/Subscriber services are classified into nine groups, some are used by the same application category. In this model, four application categories are considered: (i) Application Control, including the Signalling service; (ii) Media-Oriented, including Telephony, Broadcast Video, Multimedia Conferencing and Real-time Interactive services; (iii) Data, including the Low Priority,

Low Latency and High Throughput services; and (iv) Best Effort, including the Default Service. This ontology also includes a relationship between an SLA and a service specification, where each SLA may in fact include several different service classes, as some applications require signalling.

3.2 Service Level Specification Sections

The SLS ontology includes aggregation relations representing the following SLS sections: (i) the traffic classification section, defining the fields which identify an individual or aggregate flow; (ii) the traffic conditioning section, containing rules to identify in or out-of-profile traffic; (iii) the scope of the service, defining the boundaries of the region over which the service will be enforced; (iv) the expected QoS performance parameters; (v) the service scheduling section, defining the time period when the service is available; and (vi) the service reliability section, defining parameters related to the consistency and reliability of the service to be provided.

3.3 Network Configuration

This level of the ontology model should be observed as node centric instead of service centric as configurations are in fact performed on the nodes. In the ISP network there are two main types of nodes: edge nodes (Ingress and Egress), and core nodes. Edge nodes are far more complex as they include policers and classifiers, while core nodes just assure that the Per-Hop Behaviour (PHB) specified by the ingress is kept for each packet. As a result, the following configurations are possible: (i) configuration of the queueing disciplines; (ii) configuration of queue congestion control, and (iii) mapping classes into queues.

Associated to a node link, there may be both priority queueing and rate queueing disciplines. A priority queueing system is a combination of a set of queues and a scheduler that empties them in priority sequence. When asked for a packet, the scheduler inspects the highest priority queue, and if there is data present returns a packet from that queue. Similarly, a rate-based queueing system is a combination of a set of queues and a scheduler that empties each at a specified rate. Each queue is associated to a DiffServ Code Point (DSCP) and some queues may have congestion control through Active Queue Management (AQM) consisting of a variety of procedures that use packet dropping or marking to manage the depth of a queue.

4 Conclusion and Future Work

This work intends to go much further than a service specification. Service specifications usually include several sections, which describe the service requirements with different technical perspectives: traffic classification, traffic conditioning, scope, expected QoS, scheduling and service reliability. However, those specifications never include network configuration and the related information. In those approaches, mapping services into network configurations cannot be done

conceptually, because XML is often used and it imposes a hierarchical structure, which is not adequate to specify complex relations between services and network devices. By modeling network services in terms of an ontology, those limitations are overcome. Moreover, several classes and relations may not be explicitly defined, as they may be deduced through inference rules.

Work is currently in progress to create a service specification beyond its technical aspects, involving the administrative and management perspectives.

Acknowledgements. A PhD grant provided by *Fundação para a Ciência e Tecnologia* (SFRH/BD/17579/2004) is gratefully acknowledged.

References

1. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* **43** (1995) 907–928
2. Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 (2001)
3. Morand, P., Boucadair, M., P. Levis, R.E., Asgari, H., Griffin, D., Griem, J., Spencer, J., Trimintzios, P., Howarth, M., Wang, N., Flegkas, P., Ho, K., Georgoulas, S., Pavlou, G., Georgatsos, P., Damilatis, T.: *Mescal D1.3 - Final Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements*. Mescal Project IST-2001-37961 (2005)
4. Diaconescu, A., Antonio, S., Esposito, M., Romano, S., Potts, M.: *Cadenus D2.3 - Resource Management in SLA Networks*. Cadenus Project IST-1999-11017 (2003)
5. Alipio, P., Lima, S., Carvalho, P.: XML Service Level Specification and Validation. In: 10th IEEE Symposium on Computers and Communications (ISCC'05). (2005) 975–980
6. Goderis, D., T'joens, Y., Jacquenet, C., Memenius, G., Pavlou, G., Egan, R., Griffin, D., Georgatsos, P., Georgiadis, L., Heuven, P.V.: *Service Level Specification Semantics, Parameters, and Negotiation Requirements*. Internet-Draft, drafttequila-sls-03.txt (work in progress) (2003)
7. Brickley, D., Guha, R.: *Resource Description Framework (RDF) Schema Specification*. <http://www.w3.org/TR/rdf-schema>, W3C (1999)
8. Hendler, J., McGuinness, D.: *Darpa agent markup language*. *IEEE Intelligent Systems* **15** (2000)
9. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: *OWL Web Ontology Language Reference*. W3C (2004)
10. Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: *DAML+OIL Reference Description*. W3C (2001)
11. Babiarz, J., Chan, K., Baker, F.: *Configuration Guidelines for DiffServ Service Classes*. Internet Draft (work in progress) (2005)

Contextual Synchronization for Online Co-browsing on Peer-to-Peer Environment

Jason J. Jung

INRIA Rhône-Alpes
ZIRST 655 avenue de l'Europe, Montbonnot
38334 Saint Ismier cedex, France
jjjung@intelligent.pe.kr

Abstract. In this paper, we propose a novel synchronization method based on contextual information elicited from a group of peers for online collaborative browsing on p2p environment. Thereby, the users are semantically tracked for modeling the context about their information searching tasks. The co-browsing system embedding our proposed method was shown to improve 52.7% and 11.5% communication performance, compared to single browsing and the asynchronous system, respectively.

1 Introduction

In communication systems, synchronization can be defined as the process of making sure that two or more entities contain the same up-to-date information for consistency. More particularly, in case of online cooperation between people, we consider that it means the process of comparing the user's current context and categorizing them into the groups of similar users. We focus on detecting the moment at which the context is switched to the others.

In this study, we propose semantics-based co-browsing system on peer-to-peer (p2p) environment with two main contributions; *i*) automatic organization of the groups of like-minded users by comparing the context represented as hierarchical topic paths, and more importantly, *ii*) recognition of temporal transition of context. Thus, when a user's context is changed over time, he has to be shifted to the more relevant groups. Thereby, the contextual transitions in information searching should be detected by analyzing the sequential patterns of user browsing actions.

2 Modeling Browsing Contexts and Detecting Contextual Transitions

The semantic factors are defined to measure the various relationships between users in a group, and between web pages. The web access patterns of a user group G_a aggregated during time interval T is given by a matrix $\mathcal{W}(G_a)$ of size $U \times T$, where U is the number of users in G_a . Therefore, we can extract two kinds of browsing context; *i*) personal context ($\mathcal{PC}(u_i)$) of user u_i from each row component in P , and *ii*) group context ($\mathcal{GC}(t_j)$) at a certain moment t_j from each column components.

After labeling two arbitrary web requests based on function WD , which is referring to the web directory, we can obtain both sets of hierarchical paths of the corresponding topics. Here, lightweight ontologies (e.g., web directory) are deployed to label (more exactly, conceptualize) the web pages accessed by the corresponding users¹. Let a web page wp^i categorized to the set $\{p_m^i | p_m^i \in WD(wp^i), m \in [1, \dots, M]\}$ where M is the number of all possible topic paths.

Semantic distance δ^\diamond is $\min_{m=1, n=1}^{M, N} \frac{\min((L_m^i - L_{m,n}^C), (L_m^j - L_{m,n}^C))}{\exp(L_{m,n}^C)}$ where L_m^i , L_m^j , and $L_{m,n}^C$ are the lengths of p_m^i , p_m^j , and common part of both, respectively. It is the minimum value among all combinatorial comparisons of two sets ($|p^i| \times |p^j|$). Semantic distance is assigned in the interval $[0, 1]$, and in case of complete matching, it is 0. Exponent function in denominator is for increasing the effect of $L_{m,n}^C$.

Semantic distance matrix Δ^\diamond The adjacent wp 's aggregated by either users' accesses at a certain time point (column components in \mathcal{W} for \mathcal{GC}) or a particular user's accesses during the time interval (row components in \mathcal{W} for \mathcal{PC}) of which size is either $U \times U$ for \mathcal{GC} or $T \times T$ for \mathcal{PC} , and the diagonal elements are all zero.

Semantic distance mean μ^\diamond is given by $\mu^\diamond = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^T \Delta^\diamond(i, j)$, and means the mean value of upper triangular elements in Δ^\diamond except diagonal components. It can measure the semantic consistency of the given set of web requests.

Semantic distance deviation σ^\diamond is $\sqrt{\frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^T (\Delta^\diamond(i, j) - \mu^\diamond)^2}$. measuring the degree of dispersion of the semantic distance values from a given set of web requests.

To identify the contextual transition from the streaming web accesses of a group of users, we have to consider to compute not only the semantic factors in a given interval but also the distributions of μ^\diamond and σ^\diamond like sliding windows method. Hence, the triggering patterns from these signals are regarded as important evidence. This process should be conducted by following the objective function $\min \left[\sum_{g=1}^G \mathcal{GC}_g(t_{j+1}) - \mathcal{GC}_g(t_j) \right]$ where G is the total number of groups. It means that the temporal differences of \mathcal{GC} of each group should be minimized. This process is organized as two steps;

1. **Alarming step.** Semantic distance deviation σ^\diamond of column components in \mathcal{P} is applied to capture the contextual transitions of a particular user u_i whose personal context \mathcal{PC} is different from the corresponding group's context \mathcal{GC} . By using the objective function, alarming at time t_j can be characterized to

$$\Delta(\sigma^\diamond(\mathcal{GC})) = |\sigma^\diamond(\mathcal{GC}(t_j)) - \sigma^\diamond(\mathcal{GC}(t_{j-1}))| \geq \lambda_A \quad (1)$$

where λ_A is a threshold for alarming of \mathcal{GC} transitions. Then, a user is detected by

$$u_i = \arg_i \min \Delta'_i(\sigma^\diamond(\mathcal{GC})) \quad (2)$$

$$= \arg_i \min |\sigma^\diamond(\mathcal{GC}'_i(t_j)) - \sigma^\diamond(\mathcal{GC}'_i(t_{j-1}))| \quad (3)$$

where $\mathcal{GC}'_i = \mathcal{GC} - \{i\text{-th component}\}$. Until $\Delta'_i(\sigma^\diamond(\mathcal{GC})) \leq \lambda_A$, alarming process has to be repeated.

¹ This web directory-based labeling process is explained in [1] in detail.

2. **Confirming step.** Semantic distance mean μ^\diamond of the i -th row component in \mathcal{P} is applied to make sure that his context \mathcal{PC} is changed or not. If so, the exact time point can be detected. Similarly to the previous step, confirming the conceptual transition of user u_i can be characterized as

$$\Delta(\mu^\diamond(\mathcal{PC})) = |\mu^\diamond(\mathcal{PC}[t_1, t_{s-1}]) - \mu^\diamond(\mathcal{PC}[t_s, t_T])| \geq \lambda_C \quad (4)$$

where λ_C is the threshold for confirming of \mathcal{PC} transitions. Then, the time point s surely is the moment when \mathcal{PC} of the corresponding user is changed.

Each time the streaming web accesses within a group is stored in \mathcal{W} , the alarming step has to be fulfilled. We employ the semantic distance deviation σ^\diamond to recognize the dispersion of members in a group, rather than the group context itself. Afterward, if some users would be detected in this step, the confirming step can justify if their transitions are validated or not, because the semantic distance mean μ^\diamond is useful to measure the semantic cohesion within a certain time interval.

3 Online Co-browsing with Contextual Transition

The users detected their context transitions should be reorganized to the most relevant group. Thus, they can get all information about the group members' browsing patterns through blackboard module. User interface is simply composed of three frames for web browser, blackboard, and lists of friends/groups. On p2p environment, all users utilize homogeneous system, except to the facilities for super-peers.

Even multiple users can be the super-peer in a group, but we assume that the only one be the super-peer to monitor and control the rest of members. Thus the super-peer of g -th group \mathcal{G}_g is selected by $SuP_g = \arg_i \min \sum_{k=1, u_k \in \mathcal{G}_g}^{|\mathcal{G}_g|} |\mathcal{PC}(u_k) - \mathcal{PC}(u_i)|$ which means that the super-peer user must be in the most middle of the group as minimizing the semantic distances between other members. This selection process should be conducted when the members are changed. We can say it is similar to the adaptation process of k -nearest neighborhood (k -NN) method.

Now, we want to explain reorganization process of the users whose contextual transition has been detected. By using the objective functions and super peer selection function, the reorganization process is given by $\arg_g \min_{g=1}^G |\mathcal{PC}(SuP_g) - \mathcal{PC}(u_i)|$. where G is the total number of groups in p2p network. It searches for the super-peer whose context is closest to user u_i , because the super-peer is surely regarded as the representative context of the corresponding group context. Hence, u_i can join the most relevant group \mathcal{G}_g .

4 Experimental Results

We conducted simulations to evaluate the performance of communications on the proposed method for co-browsing system. Three groups \mathcal{G}_A , \mathcal{G}_B , and \mathcal{G}_C are organized by 30 users (each ten users in a group), and then we collected the web logs dataset by letting these users to browse the testing bed space² in the fixed personal context.

² As extended from the dataset applied in [1], it is composed of 4610 web pages labeled from ODP (open directory project, <http://www.dmoz.org>).

Firstly, in order to evaluate the detection of contextual transitions of each user, we generated 30 testing sequences, including totally 583 contextual transitions, by randomly intermixing the fragments which are randomly segmented from the web log dataset. We examined how exactly the transitions could be detected with respect to measurement recall and precision. According to F1-value, set $\frac{2R \times P}{R+P}$, we empirically uncovered the best threshold values $\lambda_A = 0.4$ and $\lambda_C = 0.6$. The threshold level of confirming step seems slightly more critical, because it is for the personal context. Overall, the average of confirming step is about 33% higher than that of alarming step.

Secondly, we evaluated the performance of communications by group organization. This proves the efficiency of online co-browsing, rather than single browsing or basic co-browsing systems. While users in \mathcal{G}_A browsed without any collaboration, \mathcal{G}_B and \mathcal{G}_C was under co-browsing. But, users in \mathcal{G}_C was only considered the contextual transitions. As the final results of three group members' browsing for three weeks, \mathcal{G}_C in online co-browsing has shown only 53% web access with helping each other according to the context. Compared with \mathcal{G}_B , our proposed method has slightly improved by 11.5%.

5 Concluding Remarks and Future Work

In this paper, we have proposed online co-browsing system of which characteristics are spatially remote and temporal synchronous. It is capable of detecting the contextual transitions of users in a group, so that they are efficiently shifted into the relevant group communications. As main contribution of this paper, most importantly, we propose tracking the contextual dynamic of the groups while co-browsing, rather than modeling the consensual context of the groups.

However, this system has still many problems that have to be dealt with in future work. We modified *Levenshtein* edit distance [2] to measure the hierarchical path-labeled web pages. In order to support more general users, we obviously consider various semantic annotation methods [3] to compare the relationships between them. Another issue is topology in p2p network. Because, as mentioned in [4], the hyperlinked environment has various topological features such as authorities and hubs, we have to think over about selection process of super-peers.

References

1. Jung, J.J.: Semantic preprocessing of web request streams for web usage mining. *Journal of Universal Computer Science* **11**(8) (2005) 1383–1396
2. Levenshtein, I.: Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory* **10**(8) (1996) 707–710
3. Jung, J.J., Lee, K.S., Park, S.B., Jo, G.S.: Efficient web browsing with semantic annotation: A case study of product images in e-commerce sites. *IEICE Transactions on Information and Systems* **E88-D**(5) (2005) 843–850
4. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46**(5) (1999) 604–632

Pedestrian Modelling: A Comparative Study Using Agent-Based Cellular Automata

Nicole Ronald and Michael Kirley

The University of Melbourne
Parkville, Victoria 3010, Australia
{naron, mkirley}@csse.unimelb.edu.au

Abstract. In this paper, we examine pedestrian population dynamics using agent-based cellular automata models. Each pedestrian is treated as an agent, mapped onto a 2-dimensional grid. The behaviour of each agent is modelled as a sequence of specific choices reflecting different levels of autonomy. Simulations of bi-directional agent movement for four behaviours in different environments (corridors of different widths with permanent blocks such as walls) are conducted in order to identify outcomes of the behaviours and recommend a strategy. The results suggest that the “lookahead” behaviour, whilst similar to the “deterministic” behaviour, was strategically the best. Little difference was found between the “floor fields” and “random walk” behaviours.

1 Introduction

Creating simulations of pedestrian behaviour is difficult [1]. Walking behaviour is largely unconscious, less constrained and consequently less predictable than vehicle traffic. However, predictions of pedestrian behaviour are required for making decisions about the design of pedestrian facilities or creating management plans for pedestrian events.

Many approaches have been developed for modelling pedestrian behaviour at both macroscopic/aggregate levels and microscopic/disaggregate levels [2]. Agent-based cellular automata (CA) represent one class of model, which encapsulate fundamental movement rules at the microscopic level. These models have been used mainly for experimenting with detailed design of pedestrian areas by modelling operational (ie. stepping) decisions. The rules developed for models of pedestrian behaviour are refined with the macroscopic behaviour in mind (such as lane formation) and are not accurately based on real-life operational behaviour [3].

In this paper, we investigate some common behavioural rules used in agent-based CA models of pedestrian behaviour and identify similarities between simple rulesets and more complex and ‘realistic’ rulesets. Each pedestrian is treated as an agent, mapped onto a 2-dimensional grid of a CA. Here, the behaviour of each agent is modelled as a sequence of specific choices reflecting different levels of autonomy. An important contribution of this work is the comparative analysis of alternative agent behaviours. We limit our investigations to: (a)

bi-directional agent movement in different environments (corridors of different widths with permanent blocks such as walls), and (b) examining the effects of pedestrian density and resulting macroscopic properties of the model. In the next section, the simulation environment is outlined, followed by a description of the behaviours modelled. In section 3, experimental results are presented. We conclude with a summary of the paper and the implications of this work.

2 Simulation Model Framework

2.1 Agent-Based CA approach

The main components in the model are the environment (CA grid) and the pedestrians (agents). This segregation between environment and agents provides the necessary flexibility to model individual pedestrian behaviour independently, and subsequently complex interactions can be captured. The environment can have global properties or cell properties. The cell properties form a field over the environment - that is, the Moore-neighbourhood cells can have an effect on a cell. In this study, we are dealing in abstract time and space, however, we could assign real-world sizes to the cell dimensions and time steps based on current pedestrian design theory. Agents occupy cells in the CA grid. They attempt to move towards their destination by selecting an unoccupied neighbouring site. Agents require a direction (either up or down) and the developer can create properties as they like (such as speed, familiarity, current cell). These properties are manipulated when the agent moves.

The simulation model has been developed in Java using object-oriented principles. It is essentially a framework, which permits the developer to create and manipulate their own environment and pedestrian properties. As such, it is similar to the geosimulation framework developed for geographic automata systems [1].

2.2 Modelling Pedestrian Behaviours

Many rules have been developed for agent-based CA models of pedestrian behaviour. In this study, we focus on four behaviours to compare their performance. Is the simple model as good as the more ‘realistic’ model?

Random walk. The simplest behaviour is a random walk. This involves randomly selecting a neighbouring existing cell (ie. one that is not blocked or outside the environment). This behaviour was included as a “base” case to compare against.

Deterministic. The next behaviour uses a deterministic rule-based agent. This agent tries to move forward. If the cell is occupied or does not exist (ie. it is outside the environment or is blocked), it tries forward right or forward left. If neither of these cells exist, it tries immediate left or immediate right.

Lookahead. This behaviour uses a “lookahead” procedure to determine whether to continue in the same lane or change lanes [4]. In short, the pedestrians look ahead a number of cells in their current lane and the lanes to their left and right. If there is someone ahead of them moving in the same direction, then the gap distance is set at the actual distance to that pedestrian. If there is someone ahead of them who is moving in the opposite direction, then the gap distance is set to half the actual distance. The pedestrian then chooses the lane with the maximal gap and ties between gap distance are broken randomly.

Floor fields. This behaviour [5] uses the concept of static and dynamic floor fields to guide the pedestrians. The static floor field consists of the distance to a specific activity or exit. The dynamic floor field consists of the number of pedestrians who have passed through the cell on the way to a specific activity of exit. Each agent has a parameter that dictates whether they follow the static floor field more (representing those who are familiar with the environment) or the dynamic floor field more (representing those unfamiliar with the environment who will follow other pedestrian going to the same destination).

This behaviour has also been developed by [6] and we partially adopted a variation from this work. In crowded situations, the agents are surrounded by other agents, therefore all cells returned a probability of 0. This meant the agent did not attempt to move, leading to total gridlock.

The crux of the behaviour is shown in equation 1, which shows the calculation of the transition probability from the current cell i to a neighbouring cell j . Contributions to the probability are the static floor field S , the dynamic floor field D , the pedestrian’s preference for moving in that direction M , whether the cell is occupied or not n (which is always 1 – unoccupied – in this model) and whether the cell exists or not e . The probabilities are normalised by the factor N .

$$p_{ij} = NM_{ij}D_{ij}S_{ij}n_j e_j \quad (1)$$

In each time step, each agent chooses their next cell. If two or more agents have chosen the same cell, the collision rules decide who moves to that cell and who stays in their current cell. The rules are based on the principles in [7], however for this set of results the friction parameter was set to 0.

3 Experiments and Discussion

3.1 Model Parameters

Two different environment sizes were used in this study. The environment was 200 cells long (L) and the width was either 15 or 20 cells (W). For some runs, a permanent blockage was introduced in the form of two doors at $L/2$. Four density values (D) were used to vary to the volume of agents (V): $D = \{0.1, 0.3, 0.5, 0.7\}$ and $V = D \times W \times L$. These densities provide conditions ranging from free-flowing to very congested. Half of the agents started at the top of the environment and half at the bottom. The agents were randomly assigned entry times between 0 and $L/2$ and a starting cell in the first row.

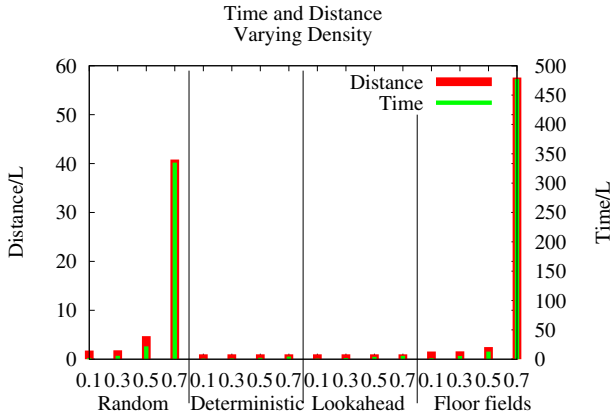


Fig. 1. Time and distance travelled divided by the length L of the environment

Movement and collision resolution was synchronous and all agents moved at most one cell per timestep. A simulation trial was run until all pedestrians reached their destination or $L \times 100$, whichever occurred first. Each scenario (environment-agent behaviour) was run several times with different random seeds.

3.2 Results

A range of simulation model outputs were recorded during each trial, including the distance covered by each agent, the time they were in the model, the number of collisions, and entropy. All results presented are an average over all trials.

The time and distance plots (Figure 1) show that the agents using the random and floor fields behaviours take longer to move through the environment. The deterministic and lookahead behaviours are more efficient.

The stops due to collisions plots (Figure 2) show that the agents using the random and floor fields behaviours are delayed due to collisions a similar amount. The deterministic behaviour performs well at low densities. However, this degrades quickly at higher densities. The lookahead behaviour appears to be approaching a maximum.

The snapshots (Figure 3) show the differences between the behaviours at the same point in time. The random and floor field behaviours cluster in the centre, whilst the deterministic agents have all moved to the right and out of each other’s way. Given their rule set, which is to move right before left, this is expected. The lookahead behaviour is also clustered, but in one large cluster. The differences between densities for the same behaviour (lookahead) are also clear. At the time for the lowest density, the agents have almost reached their destinations. At the higher densities, a cluster has formed, but agents are still moving.

In order to investigate the population dynamics in more detail, the Shannon entropy [8] of agent movement throughout a trial were recorded. This was used to compare the similarity of behaviours. An inspection of the entropy plots (Figures

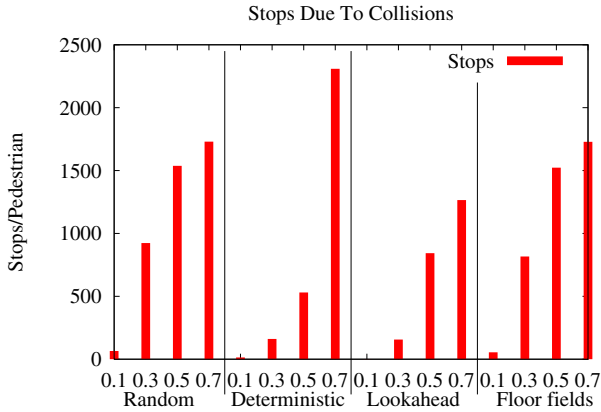


Fig. 2. The average number of times a pedestrian stops due to a collision

4 and 5) reveals that the peaks of the plots occur when the types of movement are evenly split ie. half the agents are moving and half are stopped. The troughs correspond to the situation where the agents are mostly doing the same thing ie. all moving or all stopped. The results for deterministic behaviour and the lookahead behaviour are similar, while the random and the floor fields behaviours are similar to each other. In this simple environment, the floor field behaviour is essentially an ‘intelligent’ random choice. The entropy plots also show that the behaviours go through similar phases throughout their run. Model runs with a larger number of pedestrians have similar peaks and troughs, but translated along the time axis.

For densities 30% and above, the deterministic and lookahead behaviours have a trough around $t=100$ and a peak around $t=150$ (Figure 4). At $t=100$, the agents who entered the model first have reached the centre. Up until this point, they could move freely as they were surrounded by pedestrians moving in their direction. However, at the centre they encounter pedestrians moving in the other direction and therefore collisions start occurring. At $t=150$, the two middle quarters are full of pedestrians, with very few pedestrians in the end quarters. The agents who entered the model first are encountering the last of the agents heading in the other direction. After this time, agents start to move freely again.

For the random and floor fields behaviours, there is a peak at $t=250$. As these behaviours are more likely to move sideways than the other behaviours, they take longer to reach the centre of the environment.

There are other peaks later on for some model runs. This usually occurs when a crowd has formed that has impeded a lot of pedestrians, however pedestrians are slowly making their way through. The peak occurs when there are more pedestrians free of the crowd and making their way to the exit than in the crowd still. The slight increase in width had minimal effect on the behaviour.

From the simulation results, it is clear that in this simple rectangular environment the floor field behaviour provides little advantage over the random behaviour. The floor field behaviour is more suited to a more complex situation

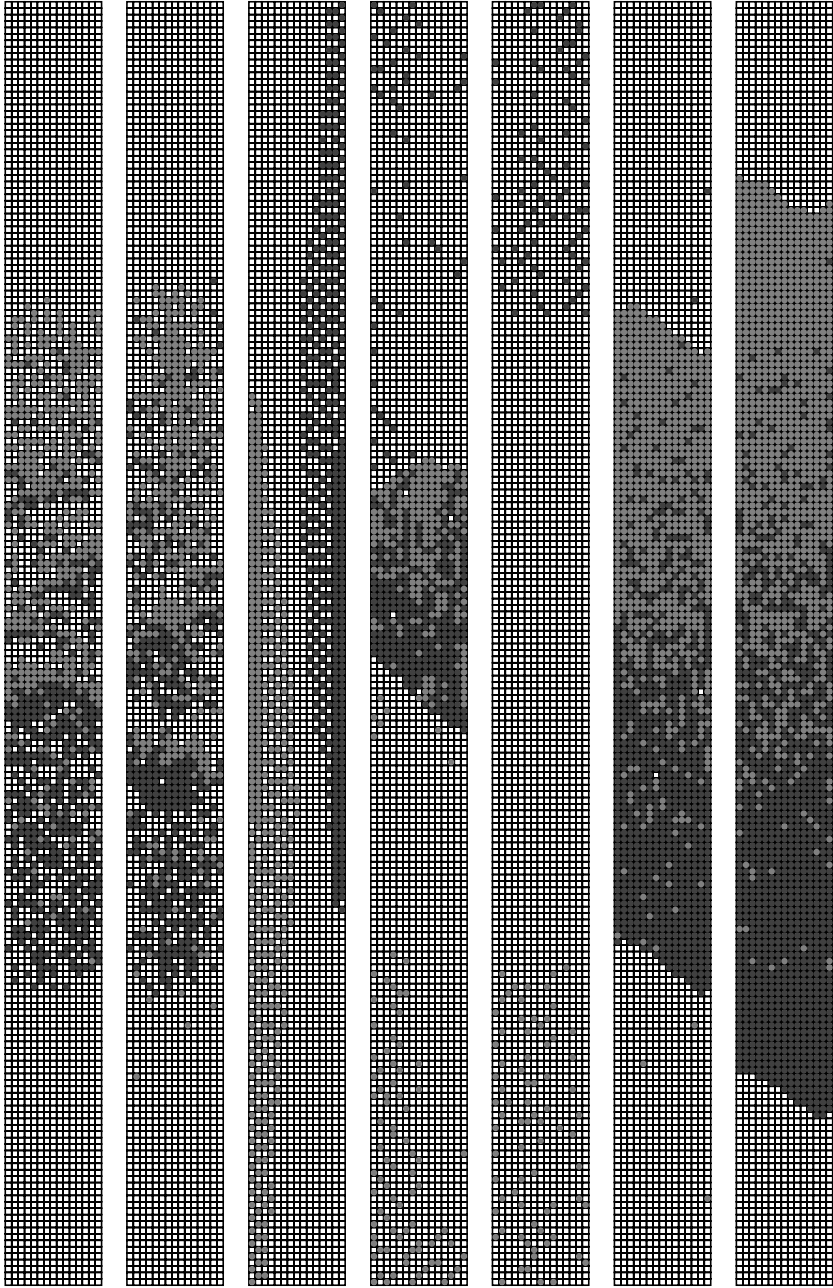


Fig. 3. Modelled crowds at the same time ($t=250$) for different behaviour and densities. The dark grey pedestrians are moving upwards, the light grey downwards. From left: random-30%, floor fields-30%, deterministic-30%, lookahead-30%, lookahead-10%, lookahead-50%, lookahead-70%.

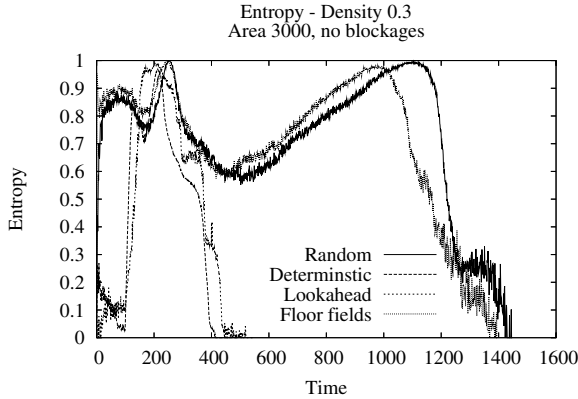


Fig. 4. Entropy plots for different behaviours with density 30%

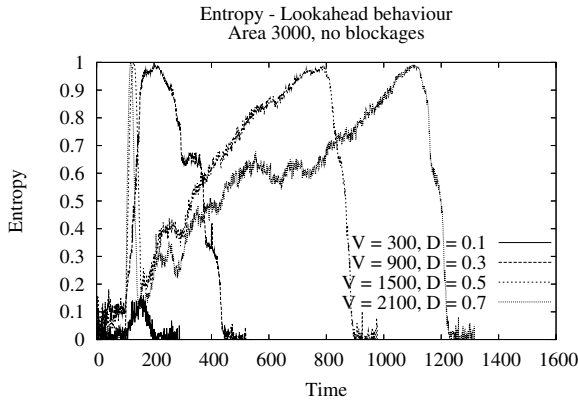


Fig. 5. Entropy plots for the lookahead behaviour at different densities

with many choices for exits and/or activities. The deterministic and lookahead behaviours were similar, however the lookahead behaviour performed better for collisions at higher densities and the differences between distance and time travelled were negligible.

4 Conclusion

The main contribution of this paper was the presentation of a comparative analysis of agent-based CA models of pedestrian behaviour of varying complexity.

Simulation experiments using different agent behaviours and environments (bidirectional corridors with varying traffic volume) suggest that the lookahead behaviour is the best strategy when moving through a simple environment, in terms of the travel time and delays. The deterministic behaviour is also a good strategy. The floor fields behaviour was the most complicated and ‘realistic’

behaviour, however the random and floor field behaviours do not perform as well in simple environments. Similarities were identified in the outputs between lookahead and deterministic and between random and floor fields, by investigating the entropy plots.

In future work, we aim to increase the complexity of the model (both the environment and range of agent behaviours). The model will be populated with agents exhibiting heterogeneous behaviours. Variable speed can also be introduced to the model. Assigning real-world values to the cell sizes and speeds and validation (obviously at a smaller scale) will also be investigated.

References

1. Benenson, I., Torrens, P.N.: *Geosimulation : automata-based modelling of urban phenomena*. John Wiley & Sons, Hoboken, NJ, USA (2004)
2. Harney, D.: Pedestrian modelling: current methods and future directions. *Road & Transport Research* **11** (2002) 2–12
3. Hoogendoorn, S., Bovy, P.H.L.: Normative pedestrian behaviour theory and modelling. In Taylor, M.A.P., ed.: *Proceedings of the 15th International Symposium on Transportation and Traffic Theory*. (2002)
4. Blue, V.J., Adler, J.L.: Cellular automata microsimulation of bi-directional pedestrian flows. *Transportation Research Record* **1678** (2000) 125–141
5. Schadschneider, A.: Cellular automaton approach to pedestrian dynamics - theory. In Schreckenberg, M., Sharma, S., eds.: *Pedestrian and Evacuation Dynamics*. Springer-Verlag, Berlin (2001)
6. Henein, C.M., White, T.: Agent-based modelling of forces in crowds. In Davidsson, P., Gasser, L., Logan, B., Takadama, K., eds.: *Multi-Agent and Multi-Agent-Based Simulation*. Volume 3415 of *Lecture Notes in Computer Science*. Springer-Verlag (2004) 173–184
7. Kirchner, A., Nishinari, K., Schadschneider, A.: Friction effects and clogging in a cellular automaton model for pedestrian dynamics. *Physical Review E* **67** (2003)
8. Boschetti, F., Prokopenko, M., Macreadie, I., Grisogono, A.M.: Defining and detecting emergence in complex networks. In Khosla, R., Howlett, R.J., Jain, L.C., eds.: *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, KES 2005*. Volume 3684 of *Lecture Notes in Computer Science*. Springer-Verlag (2005) 573–580

Nagel-Schreckenberg Model of Traffic – Study of Diversity of Car Rules

Danuta Makowiec and Wiesław Miklaszewski

Institute of Theoretical Physics and Astrophysics, Gdańsk University,
ul. Wita Stwosza 57, 80-952 Gdańsk, Poland

Abstract. The Nagel-Schreckenberg model of traffic is modified by the assumption that each car has an individual velocity limit. By simulations, the effect of supplementary rules is checked: (a) a speed limit of the slowest car is changed and/or (b) a speed limit of a car with zero gap behind is increased. It is shown that both rules increase the mean velocity; (b) rule influences the character of congested traffic – cars move though at low velocity.

1 Introduction

Recently a number of cellular automata models have been proposed in order to investigate the dynamical aspects of the traffic system [1, 2]. There are two basic cellular automata models describing single lane traffic flow: the Nagel-Schreckenberg (NaSch) model [3] and the Fukui-Ishibashi one [4]. The cellular automata traffic models, although they suffer from some limitations (see [5] for details), reproduce important real life traffic phenomena such as spontaneous formation of jams [6], explain the impact of global traffic light control strategies [7] and conditions for car accidents [8, 9]. In the framework of NaSch model, several very specific real life observations have been attempted to simulate. Let us mention the German highway rules [10], different rules for overtaking [1] or even an individual driver reaction to traffic [11].

The road network is rather poor in Poland — the network is sparse and roads are often in a bad condition. Plenty of different vehicles: bikes, horse carts, farming or construction machines and all kinds of modern cars meet together on the same road. Since usually there is no chance for overtaking, then a driver going behind slowly moving vehicles, is getting nervous and often forces the vehicle ahead to change its driving. Then usually such a hindrance vehicle (a) pulls over to let another vehicle to pass or (b) speeds-up. By this paper we implement (b) behavior by advanced NaSch model. To model car diversity we propose to violate of the common speed limit. However, when we only relax the speed limit then, obviously, a stationary state is determined by the slowest vehicles. There is a need to supplement the model by rules which influence the slowly moving cars. The extra rules proposed by us are again related to Polish drivers common habits. The first rule effects in hanging the speed limit of the slowest car. The second rule increases the speed limit of each car which distance

to a following one is 0. Here the velocity adjustment is based on the distance to a car that follows – a *looking behind rule* [12].

In the next Section we repeat the definition of the basic model, since details of NaSch model may influence results [13]. Section 3 describes the simulation procedure. We report our results concerning time development and properties of stationary states in Section 4. The final section contains summary and propositions for further model development.

2 NaSch Model

The probabilistic cellular automata model of traffic represents a lane as one-dimensional lattice of L cells. Each cell is either occupied by one of N cars or is empty. A car can move with the velocity determined by integer values bounded by a speed limit v_{max} . At time t a car is identified by a cell number $x_i^{(t)}$ occupied by the car and its velocity $v_i^{(t)}$. The number of empty cells before the car is called a gap and denoted $g_i^{(t)} = x_{i+1}^{(t)} - x_i^{(t)} - 1$. The cars move along the lane according to rules related to the driving habits:

- *acceleration*: $v_i^{(t+1)} = \min(v_i^{(t)} + 1, v_{max})$;
- *deceleration*: $v_i^{(t+1)} = \min(v_i^{(t)}, g_i^{(t)})$;
- *randomization*: if $\text{random} < p$ then $v_i^{(t+1)} = \max(v_i^{(t)} - 1, 0)$;
- *movement*: $x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$.

At each discrete time step $t \rightarrow t + 1$ the positions and velocities of all cars are synchronously updated.

The relation to empirical results is established by the notion of mean traffic flow $J = \langle v\rho \rangle$ for a given car density. In case of NaSch model, this flow depends on car density and deceleration probability $J = J(\rho, p)$. Relation between vehicle density ρ and flow J is called the fundamental diagram. At low density, $\rho \ll 1$, the flow is characterized by a linear dependence on vehicle density – a free flow state. At high densities, the flow decreases with increasing density. Vehicles are said to be in a congested state – stop-and-go waves dominate in dynamics of the system.

3 Simulation Procedure

Our simulations are performed on one-dimensional lattice of length $L = 10\,000$ sites with periodic boundary conditions. Such a system size is large enough to reduce finite-size effects [5]. If N denotes a number of cars then car density ρ is defined as $\rho = N/L$. Each simulation starts at random initial conditions: random localization of cars on a lane and random initial velocities. For each initial configuration, we update the individual vehicle velocity and position in accordance with the update rules described in the previous section. By observing first L time steps we obtain a picture of the development in time of a given system property while averaging over next L time steps we get the mean value of the

property in a stationary state. Each simulation experiment is repeated for a number (100) of different realizations.

4 Individual Velocity Limit

4.1 Definitions of Supplementary Rules

Let us assign an individual speed limit to each driver: $v_{max}(i), i = 1, \dots, N$ and $v_{max}(i) \in 1, 2, \dots, v_{lim}$. Let us denote this model as $(\mathbf{0}, \mathbf{0})$. The car flow $J(\rho, p)$ is determined by slowly moving vehicles due to the hindrance effect of other cars. It means that in a stationary state, all cars are moving in a few clusters which are led by cars with the speed limit equal to 1. Therefore we modify the model: at each time step one car chosen from of all cars driving with the smallest actual velocity, obtains a new maximal velocity at random. Let us denote this rule as $(\mathbf{1}, \mathbf{0})$.

In Fig. 1 the resulting flow is shown for all values of ρ and p simulated by us. The fundamental diagrams having the usual form, for two particular values of the randomization parameter p (important for further simulations), are presented in Fig. 2. The linear dependence of traffic flow on car density is observed in one third of the density interval. But the mean velocity of the traffic is reduced considerable, as if the basic NaSch model with $v_{max} = 1$ was simulated.

The following modifications are proposed to accelerate traffic:

$(\mathbf{2}, \mathbf{0})$: a new velocity limit, assigned to the slowest car, must be greater than the car previous limit

$(\mathbf{0}, \mathbf{1})$: if a gap between two subsequent cars is equal to 0, then the speed limit of the car ahead is increased by 1. The new speed limit cannot exceed the overall limit v_{lim} .

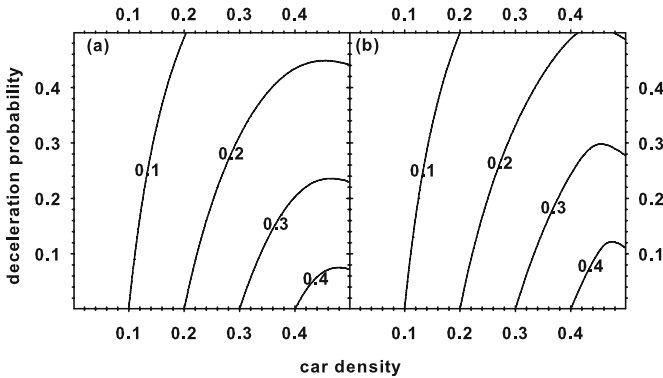


Fig. 1. Traffic flow when each vehicle has its own speed limit $v_{max}(i)$ chosen at random from $1, \dots, 10$ (left panel) and from $1, \dots, 90$ (right panel) where the slowest driving vehicle changes the speed limit to a new one — model $(\mathbf{1}, \mathbf{0})$.

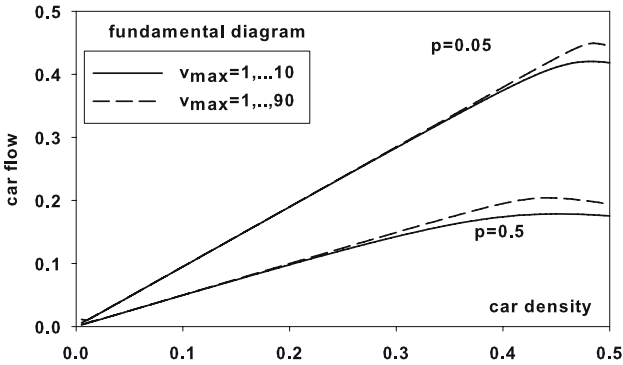


Fig. 2. The fundamental diagram for (1, 0) model in case $v_{lim} = 10$ and $v_{lim} = 90$ for $p = 0.05$ and $p = 0.5$. Note that in case of the NaSch model with $v_{max} = 1$, $\rho = 0.5$, resulting flow corresponds to maximal flow.

The combination of the above notations means that the corresponding combination of the rules is applied simultaneously. For example, (1, 1) means that the slowest driving car changes the speed limit and all vehicles, which distance to a car behind is zero, have their speed limit increased by 1.

4.2 Results

Starting from random initial states with individual velocity limits taken at random from the interval $1, 2, \dots, v_{lim}$, we observe the development of mean velocity and mean maximal velocity. The model parameters are chosen in a way which allows us to investigate different aspects of road conditions:

- (A) – basic traffic: $\rho = 0.01$ $p = 0.05$ and $v_{lim} = 10$,
- (B) – high car density: $\rho = 0.1$ $p = 0.05$ and $v_{lim} = 10$,
- (C) – high car diversity: $\rho = 0.01$ $p = 0.05$ and $v_{lim} = 90$,
- (D) – high randomization: $\rho = 0.01$ $p = 0.5$ and $v_{lim} = 10$.

The time evolution of the mean velocity and changes of the average maximal velocity are shown in Fig. 3. The time axis is logarithmic to better visualize different features of subsequent time intervals.

When the car density is low, the supplementary rules accelerate the traffic in short times what, after many time steps, lead to mean velocities higher than (0, 0) rule provides. However, this occurs to all rules except one: (1, 0) rule. After many time steps only this rule leads to the mean speed limit smaller than in (0, 0) model. In general, it appears that the mean maximal velocity increases for all rules except (1, *) rules. This is caused by our update configuration algorithm. The most left Of slowest cars on a lane is chosen to alter its sleep limit. After many time steps, cars drive with the same speed in jammed clusters. Since the most left car is often a car from a tail of a cluster, it should have a high speed limit. Therefore, a random change means here the exchange of a high speed limit to some other one, probably

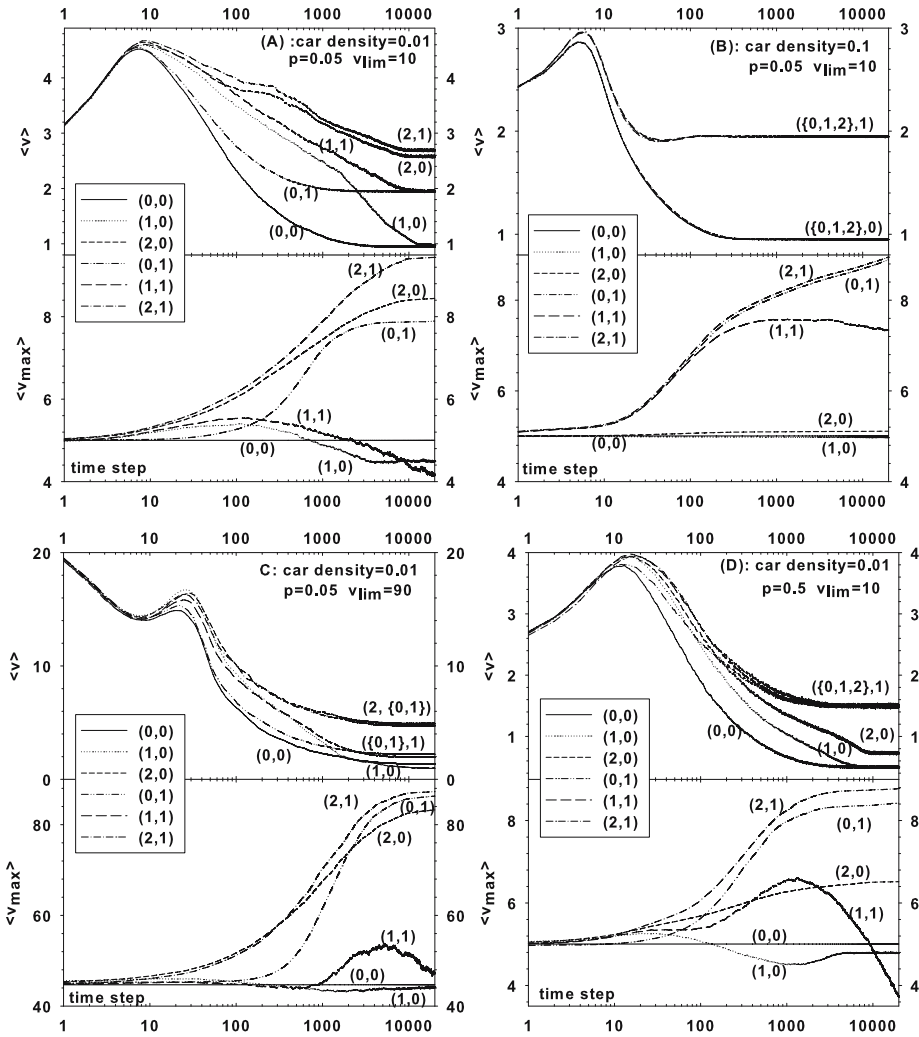


Fig. 3. Time development of the mean velocity $\langle v^{(t)} \rangle$ and mean maximal velocity $\langle v_{max}^{(t)} \rangle$ when each vehicle has its own maximal velocity $v_{max}(i)$. The label at a curve indicates the rule applied.

lower. After long evolution, the decrease of the mean speed limit is observable. Our simulations show that in the case (D) $\langle v_{max} \rangle$ does not stabilize within 10^5 time steps though the mean velocity stabilizes after 10^4 time steps.

The rules: $(*, 1)$, independently on the car density, lead to the mean traffic velocity higher than $(0, 0)$ model. When a car density is small then also $(2, 0)$ rule is efficient in establishing faster traffic.

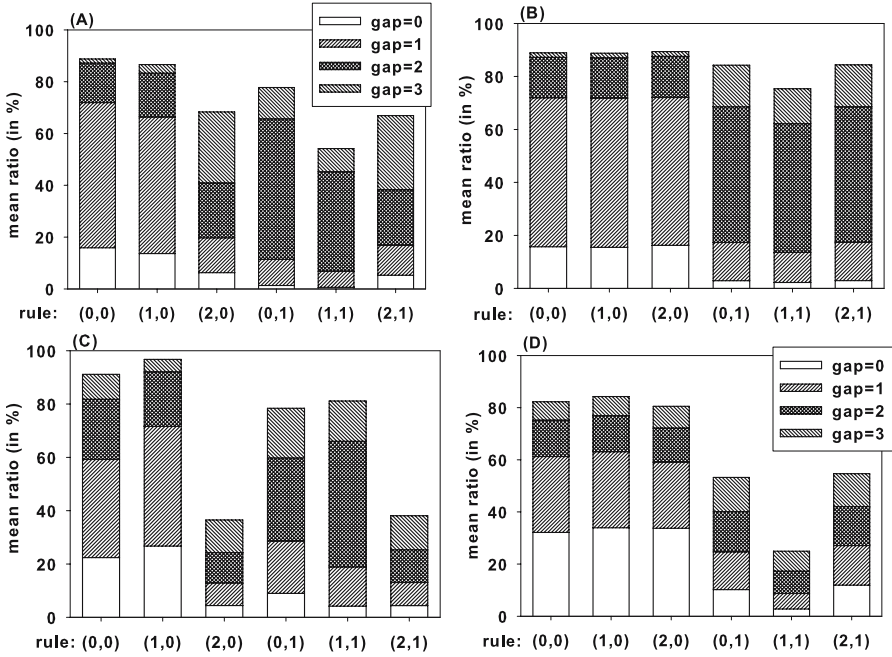


Fig. 4. Distribution of small gaps for different road conditions (A) – (D) and different rules

In all road cases: (A)–(D) and for (2, 1) and (0, 1) rules, the mean speed limits significantly increase to values close to v_{lim} . However, this high mean speed limits only little influence mean velocities. (This property is present also in the model with (2, 0) rule at low car density). Thus, although drives are allowed to go fast, they drive at about one fourth of a speed limit. To find explanation of this discrepancy we examine the structure of a stationary configuration. In particular, we concentrate on a distribution of small gaps: $g_i = 0, 1, 2, 3$. The series of plots in Figs. 4 present mean percentages of small gaps in stationary configurations.

Gaps 0 and 1 are crucial in establishing the stop-and-go state of congested traffic. The two rules: (0, 0) and (1, 0), independently on road conditions, lead to a state where more than 60% of vehicles are trapped in the stop-and-go phase; only few cars are separated more than 3 empty cells.

The rules $(*, 1)$ stabilize on configurations where stopped cars occur rarely. It means that the (0, 1) part of this rules is applied rarely. Thus there are no 2-cell clusters on a lane filled by two cars. Note that in all road condition the system stabilizes on configurations in which more than 90% cars are separated by empty cells. It means that even in case (B) of high car density, almost all vehicles are moving.

In case of high randomization (D) only about 25% of vehicles are separated by small gaps. It appears that the distribution of large gaps exhibits a power-law dependence on a gap size, see Fig. 5(a). Each time one meets a power-law

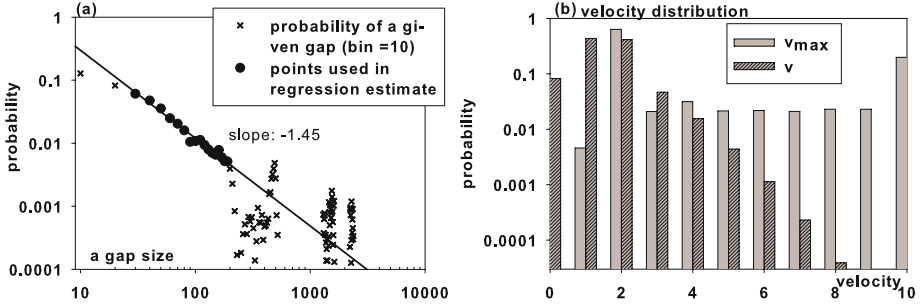


Fig. 5. (a) Log-log plot of distribution of large gaps in case of a model with (1, 1) rule and (D) parameters. (b) Histograms of actual car velocity and speed limits for a model with (1, 1) rule and (D) parameters. Subsequent bars correspond to subsequent 2000-time-step counting started after 10000 initial time steps. (log scale)

dependence, a suggestion is born that a stable configuration is critical in some sense[14]. In Fig. 5(b) a distribution of actual car velocities is confronted with a distribution of speed limits. Note that the speed limit for more than 80% of cars is 2. Hence the slow mean traffic velocity is related to the speed limit distribution. The criticality emerges from a delicate balance between effects of the parts of (1,1) rule which act oppositely to the speed limit: after many time steps, (1,0)-part works systematically on decreasing $\langle v_{max} \rangle$ while (0,1)-part at high p is effective in increasing $\langle v_{max} \rangle$.

5 Summary

In the presented paper we investigated effectiveness of rules mimicing habits popular among drivers on Polish roads. Computer experiments were performed on the base of NaSch traffic model. The car diversity was achieved by violation of the car speed limit. The first specific supplementary rules $(*, 0)$ were examples of global rules – the speed limit of a single slowest car was modified, while the second ones $(*, 1)$ were a kind of local rules – the speed limit is changed to each car which stopped another car.

Simulations exemplify the basic expected result – slowly moving vehicles determined traffic. While effect of $(*, 0)$ rules on the mean traffic velocity was small but the insistence of the car behind in speeding-up of a slow vehicle (rules: $(*, 1)$), increased the mean velocity. We also found that the character of congested traffic was different in case of $(*, 1)$ rules – the stop-and-go waves were absent. Cars were driving permanently though at low velocity.

Long runs magnified some details of our algorithms. For example, in case of (1, 1) rule we could observe emerging of properties which are characteristic for dynamical systems being in a critical state.

In future simulations, the algorithm of changing a speed limit of one of the slowest cars, should be modify in a way to avoid decrease of maximal velocity,

e.g., a car located as most right on a lane should alter its limit. On the other hand, this slowest car could be shifted to the end of a cluster behind to imitate pulling over of a hindrance to let other vehicles to pass.

Acknowledgments

We acknowledge the support of Gdańsk University – project BW-5400-5-0166-5 and Academic Computer Centre in Gdansk for CPU access.

References

1. Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. *Phys. Rep.* **329** (2000) 199–329
2. Helbing, D.: Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.* **73** (2001) 1067–1141
3. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *J. Phys. I France* **2** (1992) 2221–2229
4. Fukui, M., Ishibashi, Y.: Traffic flow in 1d cellular automaton model including cars moving with high speed. *J. Phys. Soc. Jpn.* **65** (1996) 1868–1870
5. W. Knospe, L. Santen, A.S., Schreckenberg, M.: An empirical test for cellular automaton models of traffic flow. *Phys. Rev. E* **70** (2004) 016115–1–016115–25
6. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Towards a realistic microscopic description of highway traffic. *J. Phys. A* **33** (2000) L477–L485
7. Brockfeld, E., Barlovic, R., Schadschneider, A., Schreckenberg, M.: Optimizing traffic lights in a cellular automaton model for city traffic. *Phys. Rev. E* **64** (2001) 056132
8. Boccara, N., Fukś, H., Zeng, Q.: Car accidents and number of stopped cars due to road blockage on a one-lane highway. *J. Phys. A* **30** (1997) 3329–3332
9. Moussa, N.: Car accidents in cellular automata models for one-lane traffic flow. *Phys. Rev. E* **68** (2003) 036127
10. Ebersbach, A., Schneider, J., Morgenstern, I.: Simulation traffic on german highways based on the nagel-schreckenbergs-model. *Int. J. Mod. Phys. C* **12** (2001) 1081–1089
11. Moussa, N.: Cellular automata for traffic flow with "slow-to-start" rule: Effect of randomization. *Int. J. Mod. Phys. C* **15** (2004) 29–43
12. Nakayama, A., Sugiyama, Y., Hasebe, K.: Effect of looking at the car that follows in an optimal velocity model of traffic flow. *Phys. Rev. E* **65** (2001) 016112–1–016112–6
13. Xue, Y., Dong, L., Li, L., Dai, S.: Effects of changing orders in the update rules on traffic flow. *Phys. Rev. E* **71** (2005) 026123–1–026123–6
14. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality. *Phys. Rev. A* **38** (1988) 365–374

Path-Planning for Multiple Generic-Shaped Mobile Robots with MCA

Fabio M. Marchese and Marco Dal Negro

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano - Bicocca
Via Bicocca degli Arcimboldi 8, I-20126, Milano, Italy
`fabio.marchese@disco.unimib.it`

Abstract. In this paper is described a fast Path-Planner for Multi-robot composed by mobile robots having generic shapes and sizes (user defined) and different kinematics. We have developed an algorithm that computes the shortest collision-free path for each robot, from the starting pose to the goal pose, while considering their real shapes, avoiding the collisions with the static obstacles and the other robots. It is based on a directional (anisotropic) propagation of attracting potential values in a 4D Space-Time, using a Multilayered Cellular Automata (MCA) architecture. This algorithm searches for all the optimal collision-free trajectories following the minimum valley of a potential hypersurface embedded in a 5D Time-Space.

1 Introduction

In this paper we describe a safe path-planning technique for multiple robots based on Multilayered Cellular Automata. The aim is to design a Coordinator for multi-robot systems that interacting with the environment and reacting as fast as possible to its dynamical events, decides the motions of a team of robots. Many authors have proposed different solutions during the last twenty-five years, based, for example, on a geometrical description of the environment (e.g. [10, 11]). The path-planners working on these types of models generate very precise optimal trajectories and can solve really difficult problems, also taking into account non-holonomic constraints, but they are very time consuming, too. In our opinion, to face a real dynamical world, a robot must constantly sense the world and re-plan accordingly to the new acquired information. Other authors have developed alternative approaches less precise, but more efficient: the Artificial Potential Fields Methods. In the eighties, Khatib [7] first proposed this method for the real-time collision avoidance problem in a continuous space. Jahanbin and Fallside first introduced a wave propagation algorithm in the Configuration Space (*C-Space*) on discrete maps (*Distance Transform* [6]). In [2], the authors used the Numerical Potential Field Technique on the *C-Space* to build a generalized Voronoi Diagram. Zelinsky extended the *Distance Transform* to the *Path Transform* [15]. Tzionas et al. in [13] described an algorithm for a diamond-shaped holonomic robot in a static environment, where they let a CA to build a

Voronoi Diagram. In [14] propose the coordination of robots using a discretized 3D *C-Space-Time* (2D workspace and time) for robots with the same shape (only square and circle) and a quite simple kinematics (translation only). La Valle in [8] applies the concepts of the Game Theory and multiobjective optimization to the centralized and decoupled planning. A solution in the *C-Space-Time* is proposed in [4], where the authors use a decoupled and prioritized path planning in which they repeatedly reorder the robots priorities to try to find out a solution. It can be proofed that these approaches are not complete. In this paper, we introduce a development of previous works of the authors on single robot path-planning: the multi-robot path-planning. To face with multiple robots coordination it is necessary to introduce the Time to handle precisely the space occupation of each robot in every instant. We have used CA as a formalism for merging a Grid Model of the world (Occupancy Grid) with the *C-Space-Time* of multiple robots and Numerical (Artificial) Potential Field Methods, with the purpose to give a simple and fast solution for the path-planning problem for multiple mobile robots, with generic shapes and kinematics. This method uses a directional (anisotropic) propagation of distance values between adjacent automata to build a potential hypersurface embedded in 5D space. Applying a constrained version of the descending gradient on the hypersurface, it is possible to find out all the admissible, equivalent and shortest (for a given metric of the discretized space) trajectories connecting two positions for each robot *C-Space-Time*.

2 Problem Statements

A wide variety of world models can be used to describe the interaction between an autonomous agent and its environment. One of the most important is the Configuration Space [9, 11]. The *C-Space* \mathcal{C} of a rigid body is the set of all its configurations \mathbf{q} (i.e. poses). If the robot can freely translate and rotate on a 2D surface, the *C-Space* is a 3D manifold $\mathbb{R}^2 \times \mathbf{SO}(2)$. It can be modelled using a 3D Bitmap \mathcal{GC} (*C-Space Binary Bitmap*), a regular decomposition in cells of the *C-Space*, represented by the application $\mathcal{GC} : \mathcal{C} \rightarrow \{0, 1\}$, where 0s represent non admissible configurations. The *C-Potential* is a function $\mathbf{U}(\mathbf{q})$ defined over the *C-Space* that "drives" the robot through the sequence of configuration points to reach the goal pose [2]. Let us introduce some other assumptions: 1) space topology is finite and planar; 2) the robot has a lower bound on the steering radius (non-holonomic vehicle). The latter assumption introduces important restrictions on the types of trajectories to be found.

Cellular Automata are automata distributed on the cells of a Cellular Space \mathbb{Z}^n (a regular lattice) with transition functions invariant under translation [5]: $\mathbf{f}_c(\cdot) = \mathbf{f}(\cdot), \forall c \in \mathbb{Z}^n, \mathbf{f}(\cdot) : \mathbf{Q}^{|\mathbf{A}_0|} \rightarrow \mathbf{Q}$, where \mathbf{c} is the coordinate vector identifying a cell, \mathbf{Q} is the set of states of an automaton and \mathbf{A}_0 is the set of arcs outgoing from a cell to the neighbors. The mapping between the Robot Path-Planning Problem and CA is quite simple: every cell of the *C-Space Bitmap* \mathcal{GC} is an automaton of a CA. The state of every cell contributes to build the *C-Potential* $\mathbf{U}(\mathbf{q})$ through a diffusion mechanism between neighbors. The trajectories are found following the minimum valley of the surface $\mathbf{U}(\mathbf{q})$. In

this work, we use a simple extension of the CA model: we associate a vector of attributes (state vector) to every cell. Each state vector depends on the state vectors of the cells in the neighborhood. There is a second interpretation: this is a Multilayered Cellular Automaton [1], where each layer corresponds to a subset of the state vector components. Each subset is evaluated in a single layer and depends on the same attribute of the neighbor cells in the same layer and depends also on the states of the corresponding cell and its neighbors in other layers. In the following sections, we describe each layer.

3 Multilayered Architecture

In Fig. 1 is shown the layers structure and their dependencies. There are two main layers: *Obstacles Layer* and the *Attraction Layer*. Each layer is subdivided in more sublayers: the *Obstacles L.* has 3 dimensions (2 for the workspace (X, Y) and 1 more for the time), while the *Attraction L.* has up to 5 dimensions (1 for the robots, 2 for the robots workspaces + 1 for their orientations (X, Y, θ) and 1 for the time). The *Obstacles L.* conceptually depends on the

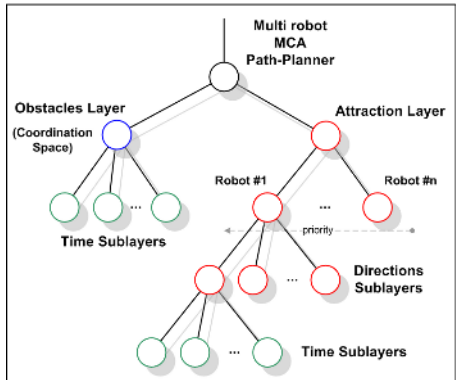


Fig. 1. MultiLayers Architecture

outside environment. Its sublayers have to react to the "external" changes: the changes of the environment, i.e. the movements of the obstacles in a dynamical world. Through a sensorial system (not described here), these changes are detected and the information is stored in *Obstacles L.* permitting the planner to replan as needed.

3.1 The Obstacles Layer

The main role of the *Obstacles Layer* is to generate a repulsive force in the obstacles to keep the robots away from them. In the present work, only static obstacles are considered (e.g. walls). For the single robot, the other robots are seen as moving obstacles, with unknown and unpredictable trajectories. We are

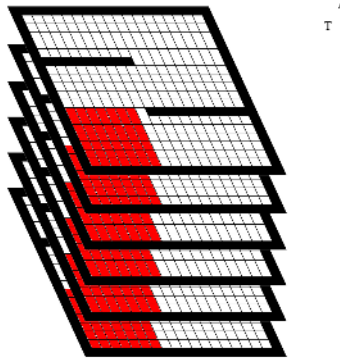


Fig. 2. Discretized C -Space-Time and C -Obstacles

considering a centralized planner/coordinator, that can decide (and, of course, it knows) the trajectories of all the supervised robots. Thanks to this knowledge, the planner considers the silhouette of the robot as an obstacle for the other robots. In this work, we introduce a discretized version of the C -Space-Time as in [14]. With the introduction of the Time axis, the robots moving in the environment become "static" obstacles in the Space-Time 4D space, having a pose in each time slice (Fig. 2). We can still call them as C -Obstacles, remembering that they are extended also in the Time dimension. In our case, we have a 3D Space-Time: $R^2 \times R$, where R^2 is the planar workspace. The time step (slice) is the time needed by a robot to move to an adjacent cell, if we consider all the robots moving at the same speed (the robots motions are synchronized).

3.2 The Attractive Layer

The *Attractive Layer* is the core of the algorithm. It computes the shortest collision-free path for a single robot, from the starting pose to the goal pose, while considering its real occupation (shape and size = silhouette). The planner is able to handle at the same time different type of robots, with different shapes, sizes and kinematics (e.g. car-like kinematics, omnidirectional, etc.). To pass from one cell of C -Space-Time to an other one, the robot can follow different paths, combining different atomic moves, such as *strict forward move*, *diagonal move*, *rotation*, and so on. Each move has its own cost (always positive); the entire set of admissible movements define the robot kinematics. The moves costs are used to build incrementally a potential surface starting from the goal cell, placed at a high time slice (the desired time of arrival), and expanding it in all the free space-time. In our case, it is a hypersurface embedded in a 4D space: $R^2 \times SO(1) \times R$, where R^2 is the workspace, enlarged with the robot orientation dimension ($SO(1)$) and the time. The goal cell receives the first value (a seed), from which a potential bowl is built adding the cost of the movement that would bring the robot from a surrounding cell to it. The calculus is performed by the automata in the space-time cell, which computes the potential value depending

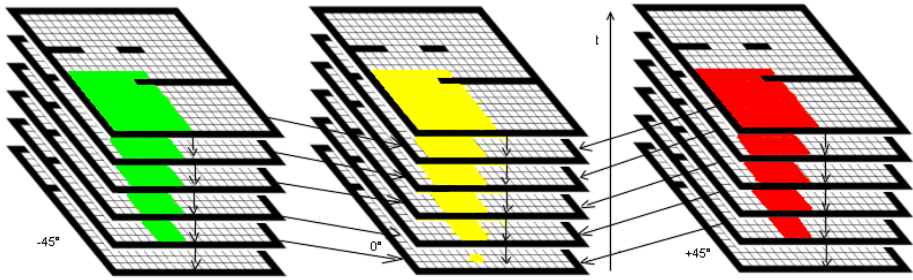


Fig. 3. Dependencies between Attractive Layers at different orientations

on the potential values of the surrounding space-time cells. Iterating this process for all the free cells, the potential surface is built leaving the goal cell with the minimum potential. The potential value has a particular mean: it is the total (minimal) cost needed to reach the goal from the current cell. Because the costs are positive, no other minimum is generated, thus avoiding the well-known problem of the "local minima". The entire trajectory is computed just following the direction of the negated gradient of the surface from the starting point. The path results to be at the minimum valleys of the surface. The robots have different goals and kinematics, hence a potential bowl for each one has to be generated separately. The potential bowl is built only in the free space, avoiding to enter in the *C-Obstacles* areas. Therefore, the robot *Attractive Layer* depends on the *Obstacles Layer*. The last one embeds the Time, thus the potential bowl varies with the time, generating different potential surfaces at different starting time and, consequently, different trajectories. In Fig. 3 is shown the dependencies between Layers and Sublayers of the overall *Attractive Layer*. Each sublayer, at a given robot orientation, depends on the adjacent sublayers at different orientations (aside in the figure), and depends also on the below temporal sublayer.

3.3 The Planning Algorithm

Entailing the layer structure previously described, is now possible to describe the algorithm to compute the robots trajectories. First of all, we have to assign the robots priorities. Up to now, no particular heuristic has been found, thus we assign them casually. The following step is to initialize the cells of the *Obstacles Layer* (full if it belongs to an obstacle, empty otherwise) in each temporal sublayer, with the obstacles distribution known from the Occupancy Map of the workspace. Then, the algorithm computes the *Attractive Layer*, based on the *Obstacles Layer*, of the robot with the highest priority level. Setting the minimum value to the goal cell, it makes the potential bowl to grow while surrounding the obstacles. With the potential surface, it extracts the shortest path, following the negated gradient from the starting cell. For each passing points of the path, then it adds the robot silhouettes, properly oriented, in each temporal sublayer of the *Obstacles Layer*. In this way, the first robot becomes a space-time "obstacle" for the lower priorities robots. The *Obstacles Layer* has a central role in this

phase, because it ensures that the robots avoid each others and the movements do not intersect, even taking into account their real extensions. For this reason it is also called Coordination Layer. The next phase is to repeat the procedure for all the other robots in order of priority. The algorithm terminates when it has computed all the robot trajectories.

4 Experimental Results

The priority planning is not complete: there are problems for which there is a “natural” simple solution, but this type of algorithm is not able to find it. In Fig. 4 (a similar one in [4]) is shown a symmetric example where adopting a priority order (e.g. the green robot passes before the red one) it does not found any solution. Fortunately, for most of the problems we do not have this kind of problems and the algorithm finds a correct solution. In the example of Fig. 5, the

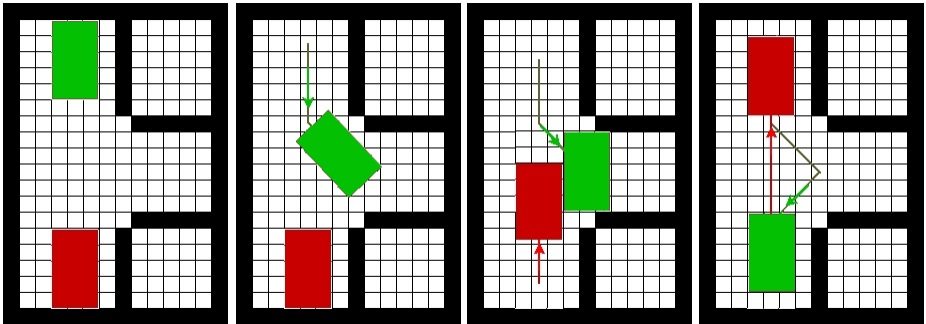


Fig. 4. Counter-example: situation for which the priority planning does not work

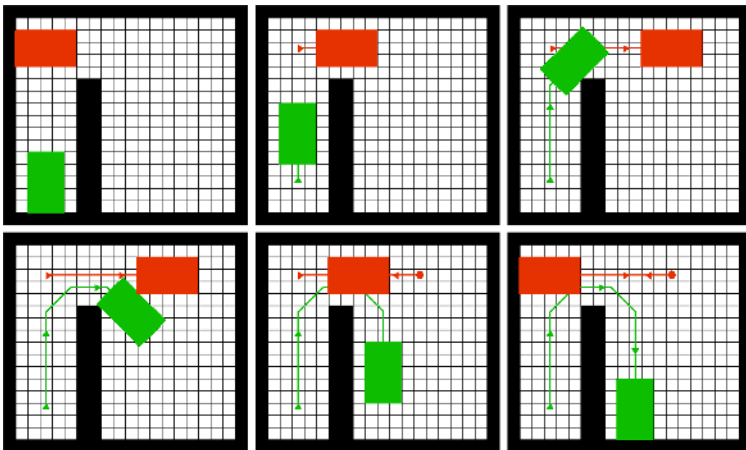


Fig. 5. Clear the way

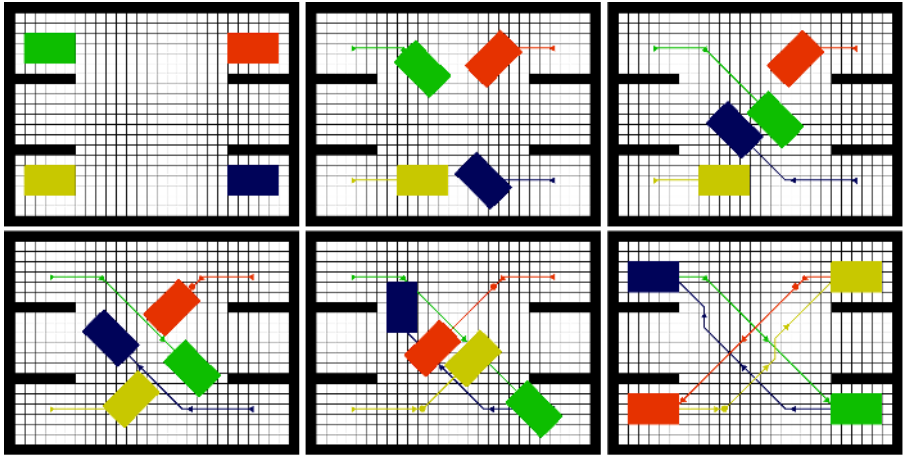


Fig. 6. Crossing robots

red robot has to clear the way to permit the green one to get out of the room, then it gets back to the original position. The following example (Fig. 6) shows a classical problem where the robots trajectories have to cross to exchange their positions in the four corners. All the robots start contemporarily, but the red and yellow robots have to stop to permit the blue and the green robots (with higher priority levels) to pass before them. Then also the red and yellow ones complete their paths.

5 Conclusions

In this paper we have proposed a decoupled and prioritized path planning algorithm for coordinating a Multi-Robot composed by mobile robots using a Multilayered Cellular Automata. One of the main topics is that we face contemporarily with mobile robots have generic shapes and sizes (user defined) and different kinematics. It is based on the Priority Planning approach in the C -Space-Time where the Time has been added to the normal C -Space.

The Priority Planning is not complete, but it works very well for most of the problems, finding all the collision-free equivalent paths for each robot. The algorithm is also able to manage the robots orientations, avoiding to waste a lot of space during the motion, and permitting to find paths in cluttered workspaces. The trajectories found are smoothed and respect the kinematics constraints and space occupancies of the robots.

References

1. Bandini S., Mauri G., Multilayered cellular automata, *Theoretical Computer Science*, 217 (1999), 99-113
2. Barraquand J., Langlois B., Latombe J. C., Numerical Potential Field Techniques for Robot Path Planning, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 22, No. 2 (Mar 1992), 224-241

3. Bennewitz M., Burgard W. Coordinating the Motions of Multiple Mobile Robots using a Probabilistic Model, *Proc. of the Int. Symp. on Intelligent Robotic Systems (SIRS)*, (2000)
4. Bennewitz M., Burgard W., Thrun S., Optimizing schedules for prioritized path planning of multi-robot systems, *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Seoul (Korea), (2001)
5. Goles E., Martinez S., Neural and Automata Networks: dynamical behavior and applications, Kluwer Academic Publishers (1990)
6. Jahanbin M. R., Fallside F., Path Planning Using a Wave Simulation Technique in the Configuration Space in Gero J. S., *Artificial Intelligence in Engineering: Robotics and Processes*, Computational Mechanics Publications (Southampton 1988)
7. Kathib O. Real-time Obstacle Avoidance for Manipulator and Mobile Robots, *Proc. of Int. Conf. on Robotics and Automation* (1985)
8. LaValle S. M., Hutchinson S. A., Optimal Motion Planning for Multiple Robots Having Independent Goals *IEEE Trans. on Robotics and Automation*, Vol. 14, No. 6 (Dec 1998), 912-925
9. Latombe J. C., Robot Motion Planning, Kluwer Academic Publishers, Boston, MA (1991)
10. Lozano-Pérez T., Wesley M. A., An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, *Comm. of the ACM*, Vol. 22, No. 10, (Oct 1979), 560-570
11. Lozano-Pérez T., Spatial Planning: A Configuration Space Approach, *IEEE Trans. on Computers*, Vol. C-32, No. 2 (Feb 1983), 108-120
12. Sipper M., Evolution of Parallel Cellular Machines - The Cellular Programming Approach, LNCS 1194, Springer (1997)
13. Tzionas P. G., Thanailakis A., Tsalides P. G., Collision-Free Path Planning for a Diamond-Shaped Robot Using Two-Dimensional Cellular Automata, *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 2 (1997), 237-250
14. Warren C., Multiple robot path coordination using artificial potential fields, *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, (1990), 500-505
15. Zelinsky A., Using Path Transforms to Guide the Search for Findpath in 2D, *Int. J. of Robotics Research*, Vol. 13, No. 4 (Aug 1994), 315-325

On Modeling and Analyzing Sparsely Networked Large-Scale Multi-agent Systems with Cellular and Graph Automata

Predrag T. Tošić

Open Systems Laboratory, Department of Computer Science,
University of Illinois at Urbana-Champaign, U.S.A.
p-tosic@cs.uiuc.edu

Abstract. Modeling, designing and analyzing large scale *multi-agent systems* (MAS) with anywhere from tens of thousands to millions of autonomous agents will require mathematical and computational theories and models substantially different from those underlying the study of small- to medium-scale MAS made of only dozens, or perhaps hundreds, of agents. In this paper, we study certain aspects of the global behavior of large ensembles of simple reactive agents. We do so by analyzing the collective dynamics of several related models of discrete complex systems based on cellular automata. We survey our recent results on dynamical properties of the complex systems of interest, and discuss some useful ways forward in modeling and analysis of large-scale MAS via appropriately modified versions of the classical cellular automata.

1 Introduction and Motivation

Multi-Agent Systems (MAS) are a research area where artificial intelligence and distributed computing overlap [28]. Hence, research in MAS heavily draws on the existing theories, tools and methodologies from both AI and distributed computing. What we would like to contribute to the more thorough understanding and better design of large-scale MAS are some ideas, paradigms and tools from another scientific discipline, namely, *complex dynamical systems*. Among many mathematical models of discrete dynamical systems, we find one class of models particularly simple yet useful for addressing many fundamental issues in distributed computing in general, and in large-scale multi-agent systems in particular. This class are the classical *cellular automata* and some of their *graph automata* extensions.

Cellular automata (CA) [10, 11, 13, 29, 30] are discrete dynamical systems whose individual components are rather simple, yet that can exhibit highly complex and unpredictable behavior due to these simple components' mutual interaction and synergy. A CA is made of a finite or infinite regular one-, two- or higher-dimensional grid of nodes, where each node behaves like a finite state machine with a fixed, usually small, number of distinct states. The nodes are interconnected together and can affect each other: the future state of a given node

depends on the current states of some of its near-by nodes. The “program” that tells a node how to update its state, based on the states of these neighboring nodes, is deterministic and fixed; it is called the *local update rule*. All nodes of a classical CA update (i) synchronously in parallel with each other, and (ii) according to the same update rule.

What are the important properties of the large-scale distributed computational and communication systems that can be adequately captured by the CA-like models? From distributed computing and multi-agent system perspectives, studying the global dynamics of a CA translates into an exploration of the global behavior of a multi-agent system when (i) the individual agent behaviors are fixed, (ii) the pattern of multi-agent interaction (“network topology”) is fixed, and (iii) both the individual agent behaviors and the interaction patterns among the agents are *homogeneous* across the entire system. However, CA-like models do not allow for an individual agent’s deliberation or adaptation of any sort: the individual agents are strictly *reactive*, and are characterized by a fixed, nonadaptive behavior. Instead, the focus of studying such models is on various *emergent dynamical properties* at the level of *agent ensembles*.

Several modifications of the basic CA model along different dimensions can be argued to provide useful abstractions for loosely coupled distributed computing systems, and, in particular, for the *large-scale* MAS made of simple reactive, autonomously executing agents. We identify the following four as the most important:

- *heterogeneity* of the cellular/graph automata in terms of (i) the individual agent behaviors and (ii) the inter-agent interaction pattern, in contrast to the strict *homogeneity* of the classical CA in both these respects;
- *model of inter-agent communication* insofar as whether the agents locally compute synchronously or asynchronously, and whether they interact with one another synchronously or asynchronously;
- *adaptability* of the individual agents, i.e., are these agents capable of *dynamically changing their behavior* via, e.g., reinforcement learning, or are their individual behaviors *fixed* once the state of their environment is specified;
- *dynamics* of the MAS network topology, i.e., whether the underlying cellular space of a cellular or graph automaton is allowed to change with time.

Among these four dimensions along which classical CA can be generalized, our recent and ongoing research has been focusing on the first two. We analyze some computational complexity implications of extending the classical CA model by allowing a limited degree of heterogeneity in individual agent behaviors, as well as in the underlying communication network, in [19, 20, 21, 23, 25]. We study some implications of the nature of inter-agent communication in [22, 24]. The purpose of the present report is to summarize some of our main findings, and to propose interpretations of those results in the context of modeling and analyzing large-scale MAS.

2 Preliminaries

Classical cellular automata are made of a finite or infinite collection of *identical* finite state machines (FSMs) that are interconnected in some highly regular fashion (e.g., [13, 29, 30]). Each copy of this FSM, called a node, updates its state based on its own current state, and the current states of the nodes in some pre-specified neighborhood, according to its *local update rule* [10, 24, 30]. All nodes update perfectly synchronously in parallel. The underlying communication network topology is called *cellular space* of a CA [10, 24]. For instance, the most studied cellular spaces in one dimension (1D) are finite rings and infinite lines of nodes.

When it comes to the very large CA, perfect synchrony of the classical CA parallel node updates is difficult to justify on either physical or distributed computing grounds. Hence, researchers have also studied various models where this perfect synchrony assumption is dropped [10, 11, 15]. For the complete formal definitions of cellular automata, their cellular spaces, and their various variants with respect to the model of inter-node interaction, we refer the reader to [24].

We next define two classes of discrete dynamical system models that result once some *heterogeneity* is allowed in the parallel and sequential CA models. More formal and elaborate definitions, as well as a detailed discussion of the motivation behind these models, can be found in our earlier work (e.g., [20, 21, 25]). *Sequential Dynamical Systems* (SDSs) are proposed in [3, 4, 5] as an abstract model for computer simulations. This model has been applied in the context of modeling and simulation of large-scale socio-technical systems; an example is the *TRANSIMS* project at the Los Alamos National Laboratory [6]. An SDS $\mathcal{S} = (G, F, \Pi)$ consists of three components. $G(V, E)$ is an undirected graph with $|V| = n$ nodes, where each node has a 1-bit state. $F = (f_1, f_2, \dots, f_n)$ is the *global map* of \mathcal{S} , with f_i denoting a Boolean function associated with node v_i . Π is a permutation of (or a total order on) the nodes in V . If the permutation Π is dropped out, and all the nodes update synchronously in parallel (the way the nodes of classical, parallel CA update their states), we arrive at the definition of *Synchronous Dynamical Systems* (SyDSs).

A *configuration* of a Boolean SDS $\mathcal{S} = (G, F, \Pi)$ or an SyDS $\mathcal{S}' = (G, F)$ is a vector $(b_1, b_2, \dots, b_n) \in \{0, 1\}^n$, where b_i is the value of the state of node v_i , for $1 \leq i \leq n$. A configuration \mathcal{C} can also be thought of as a function $\mathcal{C}: V \rightarrow \{0, 1\}^n$. A single S(y)DS transition from one configuration to another is obtained by updating the state of each node v_i using the corresponding Boolean function f_i . In case of SDSs, these node updates are carried out in the order specified by the permutation Π .

The global map computed by an S(y)DS \mathcal{S} , denoted $F = F_{\mathcal{S}}$, specifies for each configuration \mathcal{C} the next configuration $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}'$ reached by \mathcal{S} after carrying out the updates of all the node states, whether in parallel or in the order given by Π . Thus, the map $F_{\mathcal{S}}: \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a total function on the set of global configurations. This function therefore defines the dynamics of \mathcal{S} . We say that \mathcal{S} moves from a configuration \mathcal{C} to a configuration $F_{\mathcal{S}}(\mathcal{C})$ in a single transition step. Assuming that each node update function f_i is computable in

time polynomial in the size of the description of \mathcal{S} , each transition step will also take polynomial time in the size of the S(y)DS's description.

The *configuration space* (also called *phase space*) $\mathcal{P}_{\mathcal{S}}$ of an SDS or SyDS \mathcal{S} is a directed graph defined as follows. There is a vertex in $\mathcal{P}_{\mathcal{S}}$ for each global configuration of \mathcal{S} . There is a directed edge from a vertex representing configuration \mathcal{C} to that representing configuration \mathcal{C}' if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}'$. Since an SDS or SyDS is deterministic, each vertex in its phase space has the out-degree of 1.

Definition 1. A configuration \mathcal{C} of an S(y)DS \mathcal{S} is a *fixed point (FP) configuration* if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}$, that is, if the transition from \mathcal{C} is back to \mathcal{C} itself.

Definition 2. A configuration \mathcal{C} of an S(y)DS is a *cycle configuration (CC)* if there exists an integer $t \geq 2$ such that (i) $F_{\mathcal{S}}^t(\mathcal{C}) = \mathcal{C}$; and (ii) $F_{\mathcal{S}}^q(\mathcal{C}) \neq \mathcal{C}$, for any integer q , $0 < q < t$. Integer t is called the *period or length of the temporal cycle*.

Definition 3. A configuration \mathcal{C} of an S(y)DS is a *transient configuration (TC)* if \mathcal{C} is neither a fixed point nor a cycle configuration.

Thus, a fixed point is its own predecessor. A cycle configuration is reachable from itself in two or more transitions, but not in a single transition. Both FPs and CCs are *recurrent configurations*, whereas TCs are never revisited. Fixed point, cycle and transient configurations for the classical, parallel CA are defined identically to the definitions for SDSs and SyDSs. In case of arbitrary *sequential* CA, however, the above definitions need to be modified since one first has to specify what is meant by a single computational step. We refer the reader to our prior work [22, 24] for more details.

3 Parallel vs. Sequential CA: Comparison and Contrast

In this section, we summarize some of the major implications of the interaction or communication model in a multi-agent system that can be abstracted as a collection of *communicating finite state machines* (CFSMs). The two main models of interaction in CFSMs studied in the literature on cellular automata, as well as Hopfield networks [14], are (i) the perfectly synchronous, parallel node updates, and (ii) the sequential, “one-node-at-a-time” node updates. The latter model of inter-node interaction in a CA or a Hopfield network is also often referred to as *asynchronous* (e.g., [15]), although we argue in [24] that this use of terminology is somewhat inaccurate.

It has been discovered in the contexts of *discrete Hopfield networks* (DHNs), parallel and sequential CA that, for the node update rules that are required to be *linear threshold functions* [8, 27], the node updates carried out synchronously in parallel allow for a possibility of temporal cycles. In contrast, sequential node updates do not allow temporal cycles, but only transient and fixed point configurations. That is, in a *finite* CA or a DHN whose nodes update in parallel,

and according to linear threshold rules, dynamical evolution starting from an arbitrary initial configuration either converges to a temporal two-cycle or to a fixed point. In contrast, configuration spaces of the corresponding *sequential CA* and DHNs *cannot have* any CCs.

The standard proof of these two properties that contrast the parallel/synchronous and the sequential/asynchronous models of interaction among the nodes of a CA or a Hopfield network, when the number of nodes is finite, can be found, e.g., in [10]. When the linear threshold update rules are also required to be *monotone* [22, 27], we have slightly extended the fundamental result from [10] along two lines (as well as considerably simplified the original, energy function based proof) in [24]. First, we have introduced an appropriate notion of *fairness* applicable to any CFM model with sequential/asynchronous interaction among the nodes. Under the fairness assumption, we have provided a guarantee that, starting from an arbitrary initial configuration, a sequential CA (or any other graph automaton or DHN) is *guaranteed* to finitely converge to one of its *proper* fixed points [24]. Second, insofar as the finite convergence of parallel or sequential CA computations on *infinite cellular spaces* is concerned, we have characterized the circumstances under which such convergence is guaranteed on any *finite subconfiguration* of such an infinite CA. We summarize this characterization of the fundamental configuration space and convergence properties of sequential and parallel CA in the Proposition below:

Proposition 1. *Let a parallel CA or a sequential SCA be defined over a finite or infinite 1D cellular space, with an arbitrary finite rule radius $r \geq 1$. Let this cellular automaton's local update rule be a monotone linear threshold function. Let's also assume, in the sequential cases, that an appropriate fairness condition holds.¹ Then for any starting configuration $C_0 \in PS(A)$ whatsoever, and any finite subconfiguration $C \subseteq C_0$, there exists a time step $t \geq 0$ such that*

$$F^{t+2}(C) = F^t(C) \tag{1}$$

where, in the case of fair SCA, the equation (1) can be replaced with

$$F^{t+1}(C) = F^t(C) \tag{2}$$

Insofar as the practical implications of these results are concerned, assume a distributed computational infrastructure can be approximated as a ring network of communicating finite state machines. If (i) each node in this infrastructure executes autonomously and (ii) changes its local state according to a monotone linear threshold rule, and (iii) there is a finite bound on the ratios of computational speeds of different nodes, and (iv) there is a finite bound on the inter-node communication delays, then this infrastructure is guaranteed to converge to a stable or steady state after a finite amount of time. In contrast, if the nodes (i.e., agents) execute mostly autonomously but so that *global synchronization* is externally imposed on the nodes insofar as when are the individual nodes allowed

¹ See [24] for a concrete such fairness condition, that is proven to be sufficient for the assertions about sequential CA in the Proposition to hold.

to change their local states, then the global state of this system may fail to ever stabilize and may, instead, wind up endlessly oscillating between two periodic configurations.

4 On the Hardness of Counting Fixed Points of Boolean S(y)DSs

Various configuration space properties of Boolean SDSs and SyDSs, as well as the computational complexity of determining those properties, have been extensively studied since the two models were introduced in the late 1990s. Barrett, Mortveit and Reidys [3, 4, 17], as well as Laubenbacher and Pareigis [16], have investigated mathematical properties of sequential dynamical systems. Barrett et al. have studied the computational complexity of several problems about configuration spaces of S(y)DSs, such as the PREDECESSOR EXISTENCE and PERMUTATION EXISTENCE problems (see, e.g., [1]). Problems related to the existence of FIXED POINT configurations are studied in [2]. Our own subsequent work further builds on the results in [2] by addressing the problems of exact and approximate *enumeration* of various structures such as the fixed points, the predecessor configurations, and the *garden of Eden* configurations [19, 20, 21, 23, 25]. Due to space constraints, we will only summarize the main findings related to the computational complexity of enumerating FP configurations for the two restricted classes of S(y)DSs – those whose nodes update according to (i) monotone and (ii) linear threshold Boolean-valued functions.

Monotone Boolean functions, formulae and circuits [27] have been extensively studied in many areas of computer science, from machine learning to connectionist AI to VLSI design. Cellular and other graph automata with the local update rules restricted to monotone Boolean functions have also been of a considerable interest (e.g., [2, 19, 22]). The problem of counting the FPs in *monotone* Boolean SDSs and SyDSs is originally addressed in [19]. In general, counting FPs of monotone Boolean S(y)DSs either exactly or approximately is computationally intractable. This intractability holds even for the graphs that are simultaneously bipartite, planar, and very sparse *on average* [19]. An example of such graphs are the *star graphs*, in which a single central node is connected to everyone else, and each non-central node is linked only to the central node. We recall that *2CNF* stands for Boolean formulae in *Conjunctive Normal Form* such that each clause contains exactly two literals [9].

Lemma 1. *Exactly enumerating the fixed points of a monotone Boolean SDS or SyDS defined over a star graph, and such that the update rule of the central node is given as a MONOTONE 2CNF formula of size $O(|V|)$, is #P-complete.*

We now turn attention to S(y)DSs whose nodes update according to *linear threshold functions* [2, 22, 24]. In particular, we assume that each node of such an S(y)DS has its update rule given not as a Boolean formula, but as a tuple of integers that encode the appropriate weights and the threshold; for more on

some important complexity-theoretic implications of the exact encoding of the nodes' update rules, we refer the reader to our detailed discussion in [21].

Lemma 2. *Exactly enumerating the FPs of an arbitrary $S(y)DS$ all of whose nodes use Boolean linear threshold update rules with small integer weights is $\#\mathbf{P}$ -complete.*

Moreover, the computational hardness of enumerating FPs still holds even when the following additional restrictions simultaneously hold: (i) the update rules are also *monotone* (i.e., in particular, when no negative weights are allowed), (ii) the underlying graph of an $S(y)DS$ is *uniformly sparse* so that every node has only $O(1)$ neighbors, and (iii) there are only two different update rules, i.e., there are only two possible different local behaviors among the nodes.

Proposition 2. *Exactly enumerating the fixed point configurations of monotone Boolean SDSs and $SyDS$ s is $\#\mathbf{P}$ -complete, even when all of the following restrictions simultaneously hold:*

- *the monotone update rules are linear threshold functions;*
- *at most two different positive integer weights are used by each local update rule;*
- *each node has only $O(1)$ neighbors in the underlying graph of this $S(y)DS$;*
- *only two different monotone linear threshold rules are used by the $S(y)DS$'s nodes.*

5 Concluding Remarks

We propose using cellular automata (CA) and their graph automata extensions for modeling large-scale multi-agent systems made of reactive, autonomously executing agents. We focus on two particular aspects of CA. One is the issue of (a)synchrony of the node updates, as an abstraction of communication (or other forms of inter-agent interaction) among robotic, software or other kinds of autonomous agents. We analyze some important implications of (a)synchrony of the inter-agent interactions in terms of the collective dynamics of an ensemble made of simple reactive agents. In particular, we show that, if the agents update their states according to monotone linear threshold rules, then the asynchronous local updates have an advantage of guaranteeing that the global state of the entire multi-agent system would eventually stabilize.

The second issue addressed are some implications for the global system behavior when some *heterogeneity* is introduced into (i) the communication network topology and (ii) the individual agent behaviors. In that context, we show that enumerating all stable global configurations of a MAS abstracted as an appropriate graph automaton, and therefore all its possible dynamical evolutions, is computationally intractable – even if the underlying network topology is *uniformly sparse* and each agent's behavior rather simple. Thus, even a very modest amount of heterogeneity in the individual agents' behaviors, coupled with some

non-uniformity in the inter-agent interaction pattern, can make the fundamental properties of such multi-agent system's global dynamics infeasible to predict.

In summary, classical cellular automata and their appropriately tuned extensions provide a simple, yet elegant and useful setting for formally reasoning about collective dynamics of large-scale multi-agent systems made of reactive autonomous agents.

Acknowledgements. The author would like to thank Gul Agha, Harry B. Hunt, Michael Loui, Madhav Marathe and the anonymous referees. This work was supported in part by the ONR MURI Grant N00014-02-1-0715 and the NSF Grant CNS 05-09321.

References

1. C. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns. "Predecessor and Permutation Existence Problems for Sequential Dynamical Systems", Los Alamos National Laboratory Report, LA-UR-01-668, 2001
2. C. L. Barrett, H. B. Hunt, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, P. T. Tosic. "Gardens of Eden and Fixed Points in Sequential Dynamical Systems", Proc. AA DM-CCG, *Discrete Math. & Theoretical Comp. Sci.*, pp. 95-110, 2001
3. C. Barrett, H. Mortveit, and C. Reidys. "Elements of a theory of simulation II: sequential dynamical systems" *Appl. Math. & Comput.*, vol 107/2-3, pp. 121-136, 2000
4. C. Barrett, H. Mortveit and C. Reidys. "Elements of a theory of computer simulation III: equivalence of SDS", *Appl. Math. & Comput.*, vol. 122, pp. 325-340, 2001
5. C. Barrett and C. Reidys. "Elements of a theory of computer simulation I: sequential CA over random graphs" *Applied Math. & Comput.*, vol. 98, pp. 241-259, 1999
6. R.J. Beckman, et. al. "TRANSIMS: Case Study", Dallas Ft-Worth. Los Alamos National Laboratory, LA UR 97-4502, 1999
7. P. Floreen, P. Orponen. "On the Computational Complexity of Analyzing Hopfield Nets", *Complex Systems* vol. 3, pp. 577-587, 1989
8. P. Floreen, P. Orponen. "Complexity Issues in Discrete Hopfield Networks", *NeuroCOLT Technical Report Series*, NC-TR-94-009, October 1994
9. M. R. Garey and D. S. Johnson. "*Computers and Intractability: A Guide to the Theory of NP-completeness*", W. H. Freeman and Co., San Francisco, California, 1979
10. E. Goles, S. Martinez. "*Neural and Automata Networks: Dynamical Behavior and Applications*", Math. and Its Applications series (vol. 58), Kluwer, 1990
11. E. Goles, S. Martinez (eds.) "*Cellular Automata and Complex Systems*", Nonlinear Phenomena and Complex Systems series, Kluwer, 1999
12. C. Greenhill. "The Complexity of Counting Colourings and Independent Sets in Sparse Graphs and Hypergraphs", *Comput. Complexity*, vol. 9, pp. 52-72, 2000
13. H. Gutowitz (ed.). "*Cellular Automata: Theory and Experiment*", N. Holland, 1989
14. J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities", *Proc. Nat'l Academy Sci. (USA)*, vol. 79, pp. 2554-2558, 1982

15. T. E. Ingerson and R. L. Buvel. "Structure in asynchronous cellular automata", *Physica D: Nonlinear Phenomena*, vol. 10 (1-2), pp. 59-68, January 1984
16. R. Laubenbacher and B. Pareigis. "Finite Dynamical Systems", Technical report, Dept. of Mathematical Sciences, N. Mexico State Univ., Las Cruces, 2000
17. H. Mortveit, C. Reidys. "Discrete sequential dynamical systems", *Discrete Mathematics*, pp. 281-295, vol. 226, Issue 1-3, 2001
18. D. Roth. "On the Hardness of Approximate Reasoning", *Artificial Intelligence*, vol. 82, pp. 273-302, 1996
19. P. Tošić. "On Counting Fixed Point Configurations in Star Networks", APDCM Workshop within *The 19th IEEE Int'l Parallel & Distributed Processing Symp.*, 2005; in *Proc. IEEE-IPDPS '05* (CD-Rom)
20. P. Tošić. "On Complexity of Counting Fixed Point Configurations in Certain Classes of Graph Automata", *Electronic Colloquium on Computational Complexity*, ECCC-TR05-051 (revision 2), April 2005
21. P. Tošić. "Counting Fixed Point and Gardens of Eden of Sequential Dynamical Systems on Planar Bipartite Graphs", *Electronic Colloquium on Computational Complexity*, ECCC-TR05-091, August 2005
22. P. Tošić, G. Agha. "Characterizing Configuration Spaces of Simple Threshold Cellular Automata", *Proc. of the 6th Int'l Conf. on Cellular Automata for Research and Industry* (ACRI'04), Springer LNCS series, vol. 3305, pp. 861-870, 2004
23. P. Tošić, G. Agha. "On computational complexity of counting fixed points in certain classes of graph automata", *Proc. of the 4th Int'l Conf. on Unconventional Computation*, (UC'05), Springer LNCS series, vol. 3699, pp. 191-205, 2005
24. P. Tošić, G. Agha. "Parallel vs. Sequential Threshold Cellular Automata: Comparison and Contrast", session *Complex Systems Methods 1*, in *Proc. European Conference on Complex Systems* (ECCS'05), Euro. Complex Systems Society, 2005
25. P. Tošić, G. Agha. "On Computational Complexity of Predicting Dynamical Evolution of Large Agent Ensembles", *Proc. of the 3rd European Workshop on Multiagent Systems* (EUMAS'05), pp. 415-426, Flemish Academy of Sciences, 2005
26. S. Vadhan. "The Complexity of Counting in Sparse, Regular and Planar Graphs", *SIAM J. Computing*, vol. 31 (2), pp. 398-427, 2001
27. I. Wegener. "*The Complexity of Boolean Functions*", Teubner Series Comp. Sci., Wiley, 1987
28. G. Weiss (ed.), "*Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*", The MIT Press, Cambridge, Massachusetts, 1999
29. S. Wolfram. "Theory and applications of cellular automata", World Scientific, 1986
30. S. Wolfram (ed.). "Cellular Automata and Complexity (collected papers)", Addison-Wesley, 1994

Parallel Implementation of a Cellular Automaton Model for the Simulation of Laser Dynamics

J.L. Guisado¹, F. Fernández de Vega¹, F. Jiménez-Morales², and K.A. Iskra³

¹ Centro Universitario de Mérida, Universidad de Extremadura,
Sta. Teresa Jornet, 38. 06800 Mérida (Badajoz), Spain
<http://cum.unex.es/profes/profes/jlguisado>

² Departamento de Física de la Materia Condensada, Universidad de Sevilla,
P.O. Box 1065, 41080 Sevilla, Spain

³ Section Computational Science,
Faculty of Science, Universiteit van Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract. A parallel implementation for distributed-memory MIMD systems of a 2D discrete model of laser dynamics based on cellular automata is presented. The model has been implemented on a PC cluster using a message passing library. A good performance has been obtained, allowing us to run realistic simulations of laser systems in clusters of workstations, which could not be afforded on an individual machine due to the extensive runtime and memory size needed.¹

1 Introduction

In the last two decades, computational simulations based on cellular automata (CA) have been extensively used in many fields of science and technology [1]. More recently, many parallel implementations of CA models have been presented [2]. One of the reasons is that CA are intrinsic parallel systems very suitable to be easily implemented in parallel computers to carry out high performance simulations. Another reason is that in the same period parallel computer architectures have experienced a huge development and a “democratization” due to the affordability of clusters of workstations with a very good price/performance ratio. As a consequence, parallel CA simulations have been successfully applied in many fields, see for example [3, 4, 5, 6]. In addition, different software tools for the programming of CA in parallel computers (for example [7, 8]) have been introduced.

In this work, we present a parallel implementation in two dimensions of a discrete model of laser dynamics based on CA, introduced in references [9, 10, 11]. This implementation will allow us to run large size 2D simulations of the model on clusters of workstations. In addition, the 2D implementation will be useful to test the feasibility of a parallel 3D version of the model, needed to make realistic

¹ This work was partly supported by the project OPLINK (TIN2005-08818-C04-03) of Ministerio de Educacin y Ciencia (Spain).

simulations of specific laser systems, which would necessarily require a parallel implementation due to its extensive runtime and memory requirements.

The rest of the paper is organized as follows. In Section 2 the discrete model for the simulation of laser dynamics is summarized. The parallel implementation of the model is described in Section 3. In Section 4 the performance of our implementation is analyzed. Finally, the conclusions of this study are explained in Section 5.

2 Cellular Automaton Model

A laser system is modeled by a cellular automaton [9,10] defined on a two-dimensional square lattice of $N_c = L \times L$ cells with periodic boundary conditions. Two variables $a_i(t)$ and $c_i(t)$ are associated with each node of the CA. $a_i(t)$ represents the state of the electron in node i at time t : if $a_i(t) = 0$ the electron is in the laser ground state and if $a_i(t) = 1$ it is in the upper laser state. $c_i(t) \in \{0, 1, 2, \dots, M\}$ represents the number of photons in node i at time t . A large enough upper value of M is taken to avoid saturation of the system. The state variables values, which represent “bunches” of real photons and electrons, are obviously smaller than the real number of photons and electrons in the system and connected to them by a normalization constant. The *Moore neighborhood* is considered. The transition rules, which represent the different physical processes in a laser system at the microscopic level, are:

- Rule 1. Pumping: If $a_i(t) = 0$ then $a_i(t+1) = 1$ with a probability λ .
- Rule 2. Stimulated emission: If $a_i(t) = 1$ and the sum of the values of the laser photons states in the nine neighbor cells is greater than a certain threshold (1 in our model), then $c_i(t+1) = c_i(t) + 1$ and $a_i(t+1) = 0$.
- Rule 3. Photon decay: A finite life time τ_c is assigned to each photon when it is created. The photon will be destroyed τ_c time steps after it was created.
- Rule 4. Electron decay: A finite life time τ_a is assigned to each electron that is promoted from the ground level to the upper laser level. That electron will decay to the ground level again τ_a time steps after it was promoted, if it has not yet decayed by stimulated emission.

Spontaneous emission as well as thermal contributions are simulated by a continuous noise of random photons introduced at every time step in the laser mode, by making $c_i(t+1) = c_i(t) + 1$ for a small number N_n of cells ($< 0.01\%$ of total) with randomly chosen positions. As in real lasers, these random photons are responsible of the initial start-up of the laser action.

3 The Parallel Implementation

A very large automaton must be used to obtain results that quantitatively reproduce the behaviour of lasers (macroscopic systems) because a fine-grained model is needed, or in general to implement any 3D model. The runtime for a

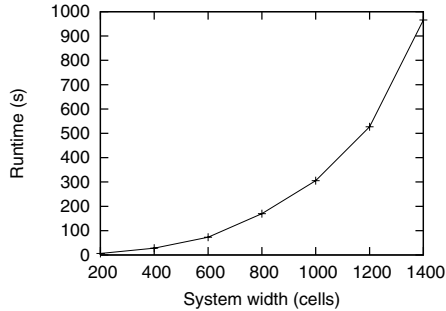


Fig. 1. Runtime of an experiment consisting in running a sequential implementation of the 2D laser model for 1000 time steps, using the same values of the parameters, for different system sizes

typical experiment grows quickly with the automaton size, as shown in Fig. 1. Therefore a parallel implementation is mandatory.

Parallelization of the model has been performed for distributed-memory MIMD (multiple-instruction multiple-data) systems using the message passing paradigm. The Parallel Virtual Machine (PVM) library has been used, because we were interested in a further study of our model using dynamic load balancing mechanisms specifically developed for this library. Nevertheless, it would be straightforward to port this implementation to other message passing libraries such as Message Passing Interface (MPI). Parallelization has been carried out following the *master-slave* programming model and the *data decomposition* or *partitioning* methodology for workload allocation: identical tasks operate on different portions of the data. A “*master program*” divides the CA grid in p partitions of equal size and sends each to a “*slave program*” running on a different processor. The particular tasks performed by the master and slave programs are:

- Master program:
 1. Input data from external file (system size, number of partitions, parameter values, number of time steps) and initialization.
 2. Spawning of slave programs.
 3. Partitioning of the initial data of the automaton.
 4. Sending of common information and initial data to each slave.
 5. Collection of results from slaves for each time step.
 6. Termination of slave programs.
 7. Calculations with the complete results.
 8. Output of final data to external files.
 9. Timing functions to measure performance.
- Slave program:
 1. Reception of common information and initial data from master.
 2. Time evolution computation for the assigned partition: application of CA evolution rules.

3. Exchange of state of the boundary cells with slave programs computing the neighboring partitions.
4. Computation of intermediate results and their communication to master program.

A diagram of this procedure is shown in Fig. 2 (a), where each box in bold type represents a different processor and the bold arrows represent the communications between the processes running on different processors.

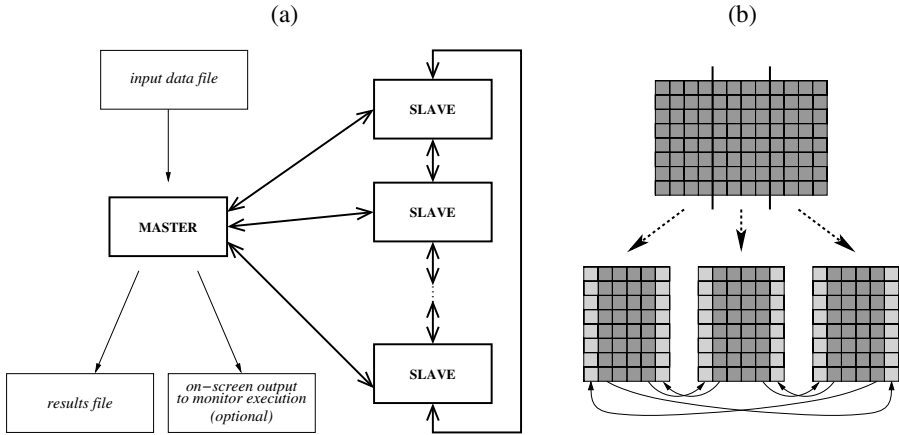


Fig. 2. (a): Block diagram of the parallel computing structure used. (b): The CA has been partitioned using a 1-dimensional domain decomposition: the automaton is vertically partitioned in stripes and each sub-domain is assigned to a different processor. Two additional columns of *ghost cells* are added at both sides of each partition to store the photon state $c_i(t)$ of neighboring cells belonging to different partitions.

In general, for d-dimensional CA 1-, 2-, ... or d-dimensional domain decompositions can be considered. Here, a 2D CA is used so that two possibilities arise: using a 1D domain decomposition, i.e. partitioning the CA in stripes, or using a 2D (checkerboard) domain decomposition. We have used a 1D domain decomposition because it makes the communication structure simpler and minimizes the number of send/receive calls (essential for an implementation using message passing). Furthermore, it is more favorable in runtime for a small to moderate number of nodes [12], despite the amount of data to be communicated is larger. As shown in Fig. 2 (b), the CA is vertically partitioned in stripes and each sub-domain is assigned to a different processor. For each sub-domain, two additional columns of ghost cells have been included at the left and right sides, used to store the photon state $c_i(t)$ of neighboring cells belonging to different sub-domains.

Each slave program computes the time evolution on its assigned partition of the automaton by applying the CA evolution rules. In each time iteration, the application of the evolution rules involves the following procedures:

1. Stimulated emission.
2. Refresh values of photon state $c_i(t)$.
3. Photon and electron decay and electron pumping.
4. Noise photons.

Procedures 1 to 3 are three successive loops through all of the cells in the partition. In the first one, stimulated emission is computed storing new values of the photon state $c_i(t)$ in a temporal array; in the second one, the values of the photon state $c_i(t)$ of all the cells in the partition are updated using the values stored in the temporal array; in the third one, photon and electron decay and electron pumping are computed. In procedure 4, N_n/p noise photons are introduced in randomly chosen cells inside the partition, where N_n is the total number of noise photons introduced in the system at every time step and p is the number of partitions in the system.

For CA in general, after each time iteration, the state of the CA cells in the boundaries of each slave partition must be communicated to the slave programs dealing with the neighboring partitions, because this state will be needed to compute the CA evolution rules there. For this particular model, the only state value from neighboring cells needed to compute the CA evolution rules for a cell is the photon state $c_i(t)$. Therefore, only this state is communicated to the neighboring partitions. This communication is carried out directly between the slave programs. In addition, in the last part of each time iteration, each slave program computes the total number of electrons $a_i(t)$ and laser photons $c_i(t)$ in its CA partition and sends this information to the master, which can record it and make some calculations with it, such as computing the Shannon's entropy as described in [9].

4 Performance Analysis

In order to test the performance of the parallel implementation, simulations have been carried out on a Linux PC cluster of ten nodes with Intel Pentium-4 processor, for different sizes of the cellular automaton grid (2520×2520 , 1260×1260 and 630×630 cells). The cluster is heterogeneous: six nodes have a clock frequency of 2.7 GHz and the other four nodes of 1.8 GHz. A particular configuration has been chosen for the experiments to avoid indeterminism in the results. For simulations with 1 to 6 nodes, the slave programs have been run on the "fast" (2.7 GHz) machines. For simulations with 7 to 10 nodes, additional "slow" (1.8 GHz) machines have been used to achieve the required number of nodes. The master program was always run on the master node of the cluster (which runs at 1.8 GHz).

The results of the simulations carried out with a sequential implementation of the 2D CA laser model (previously presented in references [9, 10]) are reproduced with the present parallel implementation. An example is shown in Fig. 3 which shows the time evolution for 1000 iterations (time steps) of the population inversion and the total number of laser photons for two different system

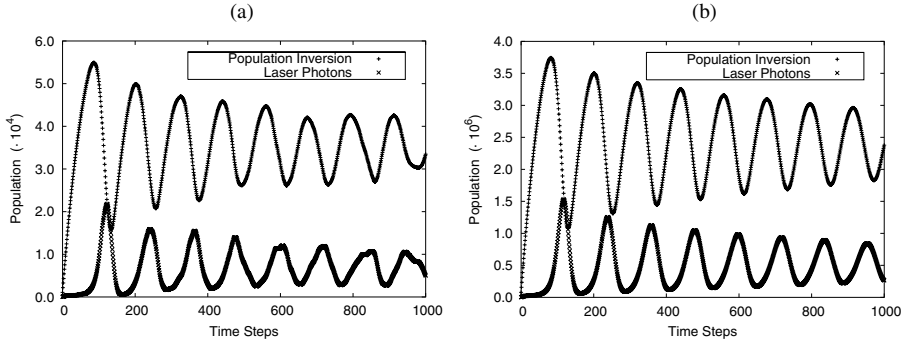


Fig. 3. Evolution of the whole system for two different system sizes. (a) (left): 300×300 cells, typical result of previous sequential implementations of the model. (b) (right): 2520×2520 cells, using the parallel implementation with a much larger system size.

sizes: 300×300 cells (a), typical system size for previous sequential implementations; and 2520×2520 cells (b). In both cases the values of the parameters are: $\lambda = 0.0125$, $\tau_c = 10$, $\tau_a = 180$. The ratio of noise photons (introduced in every time step) to total number of cells in the system has been maintained constant (0.03% of the cells) for the experiments with different system sizes. The typical laser behaviour known as *relaxation oscillations* or *laser spiking* (correlated large amplitude damping oscillations) [9] is reproduced. The differences between both results are that the populations are scaled and that the shape of the oscillations is smoother in (b) due to having used a much larger system size, thus reproducing more accurately the relaxation oscillations obtained in real lasers or from the integration of macroscopic differential equations.

The performance of the parallel implementation has been measured by running the same experiment (corresponding to the results shown in Fig. 3) using different system sizes and for different number of partitions of the whole CA (each one being handled by the slave program on a different node). The wall clock time of the experiments is shown in Fig. 4 (a), using a logarithmic scale. Runtimes get significantly shorter when the number of processors is increased, with the exception of the change from 6 to 7 processors. The reason is that “fast” machines have been used for a number of processors from 1 to 6, whereas “slow” machines are used to complete a number of processors higher than 6. As the CA operates in lock-step mode, the speed of the application is limited by the speed of the slowest running task and adding one “slow” machine results in a slower speed for the whole application.

The performance of the parallel implementation can be measured in terms of the *speedup*, ratio of the runtime of the sequential version of the program to the runtime of the parallel version. The speedup obtained for different number of partitions of the system is shown in Fig. 4 (b), in comparison with the line $y = x$, or *linear speedup*, which could be defined as the theoretical optimal speedup.

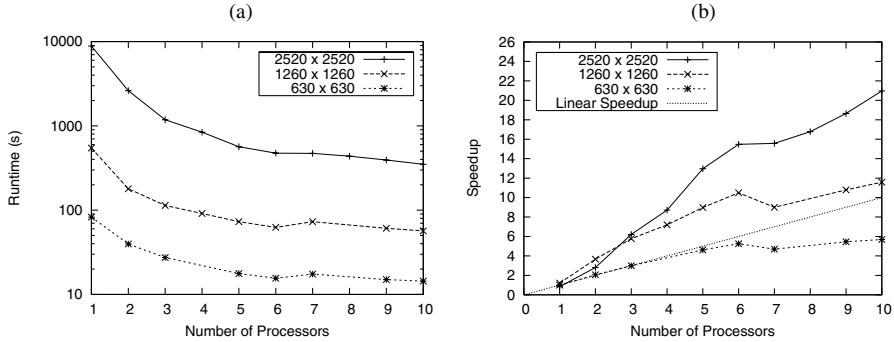


Fig. 4. (a): Runtime of the experiments, shown in a logarithmic scale, using three different system sizes, for different number of partitions of the whole CA. Each partition was run on a different processor. (b): Performance (in terms of speedup in respect to the sequential program) of the parallel implementation for varying number of processors and for three different system sizes.

A very good performance has been obtained. In fact, super-linear speedup is obtained, specially for a system size of 2520×2520 cells, and also for 1260×1260 cells. Swap memory is used for the sequential version of the program and not for the parallel version for more than one node. Therefore, a very high speedup, even higher than linear, can be obtained when comparing both magnitudes. This reason, together with similar finite memory cache effects, explains why super-linear speedup is produced. The use of swap memory for running a very large size simulation of the model (as necessary for 3D simulation) can make the calculation non-affordable on a single PC but feasible on a cluster, because the system is partitioned so that less memory is used in each node and no swap memory is needed.

In spite of the fact that, after every time step, the boundary data have to be exchanged and the intermediate results ($a_i(t)$ and $c_i(t)$) are transferred to the master program, a very good performance has been obtained. The reason is that, after the initial data are sent to all the slave processes and the computation begins, communication periods are much shorter than computation periods, as can be analyzed using a performance monitor. Therefore, the parallel application has a high computation-to-communication ratio (of the order of 10 for slave processes) and thus takes a good advantage of parallelization.

5 Conclusions and Future Prospects

A parallel implementation in 2D of a discrete model of laser dynamics using cellular automata has been presented. A CA model is an advantageous alternative to the standard description of laser dynamics, based on differential equations, when these have convergence problems—as for lasers ruled by stiff equations—or involve approximations not applicable—as for lasers with physical dimensions comparable to or even smaller than the wavelength of light, or

with an active medium of arbitrary geometry—. Another advantage of CA models is that its inherent parallel nature makes them very suitable to efficiently implement detailed simulations on parallel computers.

Parallelization has been carried out using the master-slave programming model and the data decomposition methodology for workload allocation. The parallel algorithm has been implemented for distributed-memory MIMD systems using the message passing paradigm. The implementation has been successfully run on a PC Cluster and its performance for different model sizes has been analyzed. Our implementation takes advantage of the parallelization and achieves a good speedup. This is a first step to run more realistic 3D models on clusters of workstations which could not be afforded on an individual machine due to the extensive runtime and memory size needed.

References

1. Chopard, B., Droz, M.: Cellular automata modeling of physical systems. Cambridge University Press (1998)
2. Bandini, S., Mauri, G., Serra, R.: Cellular automata: from a theoretical parallel computational model to its application to complex system. *Parallel Computing* **27**(5) (2001) 539–553
3. Sloot, P., Kaandorp, J., Hoekstra, A., Overeinder, B.: Distributed simulation with cellular automata: architecture and applications. *Lecture Notes in Computer Science* **1725** (1999) 203–248
4. Bandini, S., Magagnini, M.: Parallel processing simulation of dynamic properties of filled rubber compounds based on cellular automata. *Parallel Computing* **27**(5) (2001) 643–661
5. Dattilo, G., Spezzano, G.: Simulation of a cellular landslide model with CAMELOT on high performance computers. *Parallel Computing* **29**(10) (2003) 1403–1418
6. Love, P.J., Nekovee, M., Coveney, P.V., Chin, J., González-Segredo, N., Martin, J.M.R.: Simulations of amphiphilic fluids using mesoscale lattice-Boltzmann and lattice-gas methods. *Computer Physics Communications* **153** (2003) 340–358
7. Talia, D.: Cellular processing tools for high-performance simulation. *IEEE Computer* **33**(9) (September 2000) 44–52
8. Hecker, C., Roytenberg, D., Sack, J.R., Wang, Z.: System development for parallel cellular automata and its applications. *Fut. Gen. Comp. Sys.* **16** (1999) 235–247
9. Guisado, J.L., Jiménez-Morales, F., Guerra, J.M.: Cellular automaton model for the simulation of laser dynamics. *Physical Review E* **67**(6) (2003) 066708
10. Guisado, J.L., Jiménez-Morales, F., Guerra, J.M.: Computational simulation of laser dynamics as a cooperative phenomenon. *Physica Scripta* **T118** (2005) 148–152
11. Guisado, J.L., Jiménez-Morales, F., Guerra, J.M.: Simulation of the dynamics of pulsed pumped lasers based on cellular automata. *Lecture Notes in Computer Science* **3305** (2004) 278–285
12. Worsch, T.: Simulation of cellular automata. *Future Generation Computer Systems* **16**(2-3) (1999) 157–170

Emergent Spatial Patterns in Vegetable Population Dynamics: Towards Pattern Detection and Interpretation

Stefania Bandini, Sara Manzoni, Stefano Redaelli, and Leonardo Vanneschi

Dept. of Informatics, Systems, and Communication
University of Milan–Bicocca
Via Bicocca degli Arcimboldi 8, 20126 Milan, Italy
{bandini, manzoni, redaelli, vanneschi}@disco.unimib.it

Abstract. In this paper we present an ongoing research that aims at providing an interpretation and detection method for spatial patterns supporting ecosystem management in the study of forest systems according to a distributed modeling and simulation approach. To this aim an innovative analysis method inspired by the Chinese Go game is under design. The originality of the approach concerns the detection within system configurations of known patterns whose interpretations are well-known by expert Go players.

1 Introduction

The CAFFE (Cellular Automata For Forest Ecosystems) project is an interdisciplinary research that involves computer science, biology, and ecosystem management. It involves the Artificial Intelligence Lab (L.INT.AR.) of the Department of Informatics, Systems and Communication of the University of Milano–Bicocca and the System Research Department of Austrian Research Center (ARC). The main aim of this ongoing research is the development of methods for sustainable afforestation and management of forests. A central role is played in this project by computer supported simulations of the dynamics of forest systems. The modeling approach adopted by CAFFE for the forest system is based on Cellular Automata (see [1]) and describes the forest as the result of competition between heterogeneous vegetable populations.

In this paper, we focus on the part of the CAFFE project that aims at designing a method to support the analysis step of software simulations of vegetable populations in the forest model. This goal is particularly relevant (and ambitious) due to the distributed modeling approach that is at the basis of the forest system simulation. Analyzing the dynamics of complex systems modeled and simulated according a distributed approach is still a challenging issue in this research field: in the simulation of the dynamics of a forest composed by different species, we can have a very complex system behavior. A very interesting, and difficult, type of simulation analysis may aim at recognizing collective emergent behaviors occurring during the system simulation. Most of the available

approaches to analyze the dynamics of complex systems are based on statistics and probability theory and they aim at deriving macro level interpretations by aggregating and correlating variables of the micro level(s). Within the context of forest ecosystems for instance, the dynamics of the forest (e.g. biomass) is computed aggregating the features of living trees taking into account different age and dimensions of trees [2]. Another common approach to complex system analysis concerns the detection and interpretation of recurring patterns [3]. According to the latter approach, in this paper we propose a method to interpret detected spatial patterns that can emerge in system configuration when it is adopted a modeling approach based on Cellular Automata (CA). The originality and peculiarity of our proposal is its inspiration by Chinese Go game.

Go game, due to the simplicity of its playing rules but also to the complexity of possible configurations and the consequent complexity of playing tactics and strategies, has inspired several models (e.g. in economy, military, art, semiology, culture, and many others [4]). In Go game two populations (i.e. black and white stones) compete for survival in a territory with limited space. During the game, black and white stones situated on the Go board cannot move, but they can be put onto a board site, survive or die as result of a metaphorical competition on the local territory with neighboring stones. A key concept for the death or the survival of Go pieces is the notion of *liberty*: if a group of pieces has no liberties (i.e. none of its elements is adjacent to a free site), it is removed from the board.

As first step of this research, we have studied a subset of the patterns that can be observed on the Go board during the game and we have applied their interpretations to study the dynamics of a forest ecosystem modeled according to CA approach. As in Go game, in the CA-based model of forests, trees of different species live in a territory, compete for limited resources. The concept of liberty for plants can refer to favorable survival conditions and free available space for reproduction. Starting from these, and other, analogies (see Section 3), between Go rules and the behavior of entities of the CA-based forest model, we propose to experiment the adoption of Go interpretation of Go board configurations to analyze the dynamics of the forest system.

After a summary of the CA-based model of forest previously presented in [1], in Section 3 we introduce a subset of Go patterns and we show how their interpretation can be applied to forest ecosystems. Moreover, in Section 4 we introduce an overview and some first ideas about pattern detection problem. The paper ends with future directions for the research.

2 The CA-Based Forest Model

In this section we give an overview of the CA-based model of forests. To introduce the model, in the following we report cell state and update rule, and the CA initial configuration.

The Cells: Each cell of the automaton represents a square portion of terrain that contains some resources (i.e. water, light, nitrogen, and potassium) and can host a tree.

The finite set Q of cell states can be defined by:

$$Q = \{\mathbf{R}, \mathbf{M}, \mathbf{P}, T, \mathbf{Z}_T, \mathbf{N}_T, \mathbf{U}_T^G, \mathbf{U}_T^S, \mathbf{R}_T, \mathbf{M}_T, \mathbf{G}_T, \mathbf{S}\}$$

where \mathbf{R} is a vector referring to the amount of resources present in the cell. \mathbf{M} and \mathbf{P} indicate respectively the maximum amount of each resource and the amount of each resource produced by the cell at each update step. T is a flag indicating whether a tree is present in the cell, while $\mathbf{Z}_T = \{z_T^r, z_T^t, z_T^l, z_T^f\}$ is a vector defining the size of the different parts of the tree (in our model, roots, trunk, leaves, and fruits). \mathbf{N}_T , \mathbf{U}_T^G , \mathbf{U}_T^S and \mathbf{R}_T are vectors related to the amount of resources needed or used by the tree living in the cell. The first one defines the amounts of each resource the tree takes from the cell. The second and the third ones define the amount of each resource needed at each update step by the tree to *grow* and the amount of each resource needed to *survive*. \mathbf{R}_T defines the amount of each resource stored by the tree at previous update steps. \mathbf{M}_T is a vector of threshold values for different parameters defining the tree, such as maximum size, maximum age, minimum age for reproduction, maximum number of seeds produced for each mass unity of fruits, and so on. $\mathbf{G}_T = \{g_T^r, g_T^t, g_T^l, g_T^f\}$ is a vector defining the *growth rate* of each of the parts of the tree when enough resources are available. $\mathbf{S} = \{s_1, \dots, s_l\}$ is a vector defining the number of seeds present in the cell for each of the species growing in the territory.

The Update Rule: At each update step of the automaton, the tree present in each cell (if any) takes the resources it needs from the cell itself and uses them to survive, grow (if enough resources are available), and produce seeds. If a tree is present in cell $C(i, j)$ ($C(i, j)$ is the cell located at position (i, j) in the lattice), it takes from it a given quantity (defined by $\mathbf{N}_T(i, j)$) of each available resource $\mathbf{R}(i, j)$. The amount of resources taken depends on the size of the tree $\mathbf{Z}_T(i, j)$. If the resources available in the cell exceed its needs, the tree stores some resources adding them to vector $\mathbf{R}_T(i, j)$. Conversely, if the resources available in the cell are not sufficient, the tree uses resources stored at previous update steps. If also the resources stored are not sufficient for the tree to survive, according to $\mathbf{U}_T^S(i, j)$, the tree dies. The tree also dies when it reaches its maximum age defined in vector $\mathbf{M}_T(i, j)$. A tree can also produce some seeds, according to its age and to the size of its fruits ($z_T^f(i, j)$), as a consequence the seed vector $\mathbf{S}(i, j)$ is updated. A newborn plant can sprout in a vacant cell, if the latter contains a seed of its specie, and again enough resources. If seeds from different species are present in the cell, the winning specie is chosen at random, with probability proportional to the number of its seeds. Moreover, we defined the update rule in order to reproduce the increasing influence that a growing tree can have on neighboring cells. For example, its roots can extend beyond the limits of the cell hosting it. Or, when it gets taller, it shades an increasingly wider area around itself, thus having a negative influence on the growth of other trees in its neighborhood. We modeled the impact of a tree in a given position on its neighborhood by making resources flow from richer cells to poorer ones. In other words, a cell hosting a large tree is poor on resources, since the tree at each update step takes most (or all) of them. If the neighboring cells are vacant,

their resources remain unused, and thus are richer than the one hosting the tree. Therefore, if we let resources flow from richer cells to poorer neighbors, the effect is that in practice a large tree starts to collect resources also from neighboring cells.

Let $r_h(i, j)$ be the amount of resource h contained by cell $C(i, j)$, and assume that we are using the von Neumann neighborhood. $r'_h(i, j)$, the amount of resource i after this update sub-step, is defined as:

$$r'_h(i, j) = \frac{r_h(i, j) + \frac{r_h(i+1, j) + r_h(i-1, j) + r_h(i, j+1) + r_h(i, j-1)}{4}}{2}$$

In other words, we can see each cell as divided in four parts, each one containing the amount $r_h(i, j)/4$ of resource h , and corresponding to one of the neighbors. The amount of resource h contained in each part is balanced with the corresponding part of the neighbors. In case we adopt the Moore neighborhood, we can imagine the cells as split into eight portions.

The CA Initial Configuration: The initial configuration of the CA can be defined by the user, by setting appropriate resource parameters for each cell. Also, some trees might be already present on the territory, with all the variables defining their type. Otherwise the territory might be empty with the only presence of some seeds. According to the type of territory interested by the simulation, several types of soils and available resources for trees growth and reproduction can be set.

3 The Go-Based Pattern Interpretation Method

The analogies between Go game and the CA-based model of forests above sketched are going to be formalized. Very broadly in the CA-based model of forests, trees of different species live in a territory, compete for limited territory resources, can be born, die, and they cannot change their position. Moreover, the concept of liberty for plants can refer to favorable conditions for growth and reproduction. At the same time a Go board represents a territory divided in cells, and it can be viewed as a CA with three possible states for each cell: void, black or white (if there is a Go piece of the black or white team). Starting from these analogies, our proposal suggests to exploit this game to study emergent patterns in the dynamics of complex systems, by studying some spatial patterns well-known by advanced Go players [5], and to verify whether their interpretation can be suitably and fruitfully applied to interpret similar spatial patterns occurring in dynamics configurations of the CA-based forest model.

We introduce now some common spatial patterns that can emerge during Go games and that are well-known by Go players. Each spatial pattern is interpreted by Go players in terms of game competition, and we briefly describe how the interpretation of Go patterns can be applied to interpret spatial patterns that emerge from the evolution of the CA-based model of forests. This interpretation is validated by preliminary but encouraging results of experiments conducted

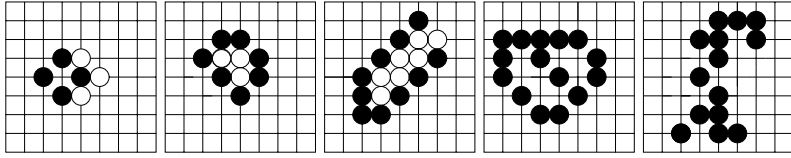


Fig. 1. Patterns in Go game: Ko, Geta, Shicho, Iki, Tsugi (from the left to the right)

on a CA-based model of the Italian Alps ecosystem (not reported here due to the lack of space but presented in [6]). An example of the Go game patterns is shown in Fig. 1.

Ko pattern: Ko is the configuration in Go game where pieces of different teams compete on a little free territory (i.e. a set of positions not occupied by stones), and no one of the two adversaries is able to take control of the territory in a stable way. In the performed simulations with the CA-based forest model, we observed similar unstable situations in which two or more tree species compete for a little territory without stable dominance. The occurrence of Ko pattern can be useful in the detection of areas in which there are a good equilibrium between the living species.

Geta pattern: Geta pattern corresponds to the local capture of a group of adversary pieces by a set of stones that surrounds it. Similarly if a group of trees is surrounded by plants of another specie and it is forced to be limited within a little and close territory zone, sooner or later it will die. In fact, trees and plants that are forced in a little zone have little space to harvest the needed resources for survival and reproduction. The occurrence of Geta pattern can be a good sign of the presence of a dominant specie in the given area.

Shicho pattern: Shicho is a pattern in which a group of Go stones expands itself towards another side of the Go board. Shicho does not imply the movement of single stones that are part of the moving group. Group movement occurs in time-space (i.e. group movement is the result of stones removal from some positions and the positioning of others in adjacent ones). In forests Shicho occurs when an homogeneous group of plants is situated in a zone where resources are not suitable to satisfy their needs. Since new born trees are more likely to survive if they grow up on areas rich of resources, also in this context, we can observe as emergent phenomenon, a group shifting in space-time. The occurrence of Shicho pattern may indicate a soil impoverishment or the rising up of a new dominant specie in the area.

Iki pattern: Iki is a strong pattern that can not be captured by the adversary. Iki corresponds to a part of Go board surrounded by stones of the same color, with some free positions in its inner side to form two ‘eyes’, in Go jargon. This formation has two internal liberties that, according to Go playing rules, cannot be occupied by adversary pieces. A spatial pattern in forest dynamics similar to Iki is characterized by a strong group that can survive for long time assuring part of the territory to its individuals. In fact, trees of this group have vacant cells in

their neighborhood and this guarantees the availability of space and resources for their survival and reproduction. The occurrence of Iki pattern shows the stability of the involved specie in the given area.

The Tsugi pattern: Tsugi is a Japanese word that means ‘connection’. Connections are very important in Go competition, because two stones connected to form a group are stronger than they alone. In fact, it is more difficult for the adversary to build a group able to surround connected stones (i.e. it may require a lot of stones). Tsugi phenomenon occurs in a natural way also in forests. Each group of plants expands itself by reproduction, and when two groups expand toward one another, there is the possibility to create a connection between them. As in Go game, two connected groups of plants are stronger because they can support each other. When a plant dies neighboring ones can replace it, and when two groups are connected, neighbors increase in number. Beside this aspect, an isolated little group of trees can easily fall in the Geta phenomenon but if it is connected to another group, it is no more possible to surround it. Tsugi pattern occurs in particular in a situation of a new territory colonization or during a repopulation phase of a given area. In the both cases it indicates an unstable situation.

All these patterns can be used for a meaningful interpretation of important phenomena in some simulation scenarios. In particular they can be useful, for example, in the case of artificial repopulation of forest in a given area with the introduction of new species. In this case it is important to understand if the new specie can survive and what are the reactions of the other living species. But while the occurrence of Ko phenomenon means a good equilibrium between species, the frequency of Iki and Tsugi indicates the formation of a strong dominance presence in the area, and Geta and Shicho mean the disadvantage of a specie in comparison with the others. All these considerations can support the decision maker in the illustrated domain problem.

A central issue to be considered in developing a method for pattern interpretation concerns the development of a method to support the detection of emergent patterns during system dynamics. This task can be very difficult for a human operator that analyzes the dynamics of the CA-based forest model, in particular for large scenarios involving several tree species. Therefore the main future work that starts from first positive results on Go-based pattern interpretation concerns the development of a detection method capable of recognizing known patterns. In the following section we report an overview and some first ideas about pattern detection problem.

4 Towards a Pattern Detection Method

In its most general sense, pattern detection can be defined as “the extraction of consistent information from (possibly noisy) spatiotemporal data”. Alternatively, it can be defined as “the act of taking in raw data and taking an action based on the category of the data” [7]. As such, it can be accomplished by a large set of many different machine learning methods. Pattern recognition aims to classify

data (patterns) based on either a priori knowledge (supervised learning) or on statistical information extracted from the patterns (unsupervised learning). In general, a complete pattern recognition system consists of (1) a sensor that gathers the observations to be classified or described; (2) a feature extraction mechanism that computes numeric or symbolic information from the observations; (3) a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features. In our applications, the observations are already given by the structure of the cellular grid at a given time step. Thus, step (1) is not necessary. Many Artificial Intelligence and statistical techniques have been developed for feature extraction (point (2)). Some examples are: discriminant analysis, principal component analysis, principal curve, factor analysis, independent component analysis, some forms of clustering and feature selection. All these techniques may be useful for our application, particularly in presence of large cellular grids. We are particularly interested in discriminant analysis, where new basic features may be considered as the current states of the grid cells and new feature may be constructed (eventually replacing the old ones) via linear or non-linear combinations, so that classification (point (3)) is easier. The choice of the classification scheme (point (3)) for pattern detection in our application is probably the most delicate: a large number of machine learning techniques have been used with success in many pattern detection applications which have similarities with ours. For example, in [8], Ripley introduces the use of many different Neural Networks models for pattern detection in many applications, such as classifying galaxies by shape, identifying fingerprints, highlighting potential tumors on a mammogram, handwriting recognition. While the second, third and fourth of these applications can be considered “simpler” than ours, the first one has some similar characteristics, in particular for the fact that galaxies are modelled as complex systems, in a similar way as we do for forests. In [9], Babuskame presented a way to solve some pattern recognition applications by means of fuzzy systems. Fuzzy rules may provide more flexibility and decisional power to the system. In [10] contains many contributions on the use of evolutionary techniques (like Genetic Algorithms) for solving many pattern detection applications. The contribution given by Wah *et al.* in this book (chapter 5) is particularly interesting, because it deals with the generalization of learned rules in genetic-based learning. Furthermore, chapters 9 to 13 of this book present some hybrid techniques integrating Fuzzy, Neural and Genetic approach. Given the complexity of our application, we are particularly interested in this last kind of system. Fuzzy, Neural and Genetic systems, in fact, are characterized by a high degree of diversity. Hybrid pattern recognition schemes have been proposed in order to combine their strengths and avoid their weaknesses. Some good discussions about this issue are contained in [11], where several techniques are combined into single pattern recognition systems. Hybrid methods have been known about for a long time, but they have gained new interest only recently. Nevertheless, in our opinion, they have never been applied to complex systems like detection of spatial patterns in vegetable population and forest dynamics.

This research field looks promising and interesting and we plan to investigate it in the future.

5 Conclusions and Future Works

In this paper we have proposed a method for pattern interpretation of emergent phenomena that aims at supporting ecosystem management in the study of forest systems by simulation. This proposal has been validated by preliminary but encouraging results of experiments conducted on the CA-based model applied to an area of Italian Alps ecosystem. These experiments were conducted to verify the correct evolution of Go-like patterns with reference to the forest simulation scenario (their positive results are available in [6]). A deeper experimentation campaign is under development in collaboration with ARC's domain experts in order to apply the presented approach on real data scenarios.

The main future work that starts from these results concerns the development of a detection method capable of recognizing and interpreting known patterns during system dynamics (first ideas have been introduced in Section 4). Another important future development of this research can be the study of new patterns for the interpretation of other phenomena obtained by the introduction of other system elements in the model (e.g. urbanization or pollution as effect of the human presence, or desertification as effect of other natural interactions).

References

1. Bandini, S., Pavesi, G.: "Simulation of vegetable populations dynamics based on cellular automata" In Bandini, S., Chopard, B., Tomassini, M., eds.: *Cellular Automata*, Volume 2493 of LNCS, Berlin, Springer-Verlag (2002)
2. D.G.Green, "Modelling plants in landscape", in *Plants to Ecosystem - Harek T. Michalewicz, ed. CSIRO, Lollingwood Ans.*, 1997.
3. S.Wolfram, "Cellular automata as models of complexity", *Nature*, 311:419-424, 1984.
4. Reysset, P.: *Le Go: aux sources de l'avenir*. Chiron (1994)
5. Soletti, G.: Note di Go. FIGG (Federazione Italiana Giuoco Go). Available for download at www.figg.org
6. Bandini, S., Manzoni, S., Redaelli, S.: "Toward the Interpretation of Emergent Spatial Patterns through GO Game: The Case of Forest Population Dynamics" In proceedings of Simulation and Formal Analysis of Complex Systems (WOA 2005)
7. R. O. Duda, P. E. Hart and D. G. Stork. *Pattern classification (second edition)*. Wiley. New York. 2001.
8. B. D. Ripley *Pattern Recognition and Neural Networks*. Cambridge University Press. 1996.
9. R. Babuska. "Fuzzy clustering algorithms with applications to rule extraction". In P.S. Szczepaniak and P.J.G. Lisboa, editors, *Fuzzy Systems in Medicine*, pages 139-173. Springer-Verlag, Heidelberg, 2000.
10. S. K. Pal and P. P. Wang *Genetic Algorithms and Pattern Recognition*. Boca Raton, FL: CRC Press. 1996.
11. H. Bunke and A. Kandel. *Hybrid methods in pattern recognition*. World Scientific Series in Machine Perception and Artificial Intelligence, Vol. 47. 2002.

Automata Network Simulator Applied to the Epidemiology of Urban Dengue Fever

Henrique F. Gagliardi^{1,3}, Fabrício A.B. da Silva³, and Domingos Alves^{1,2}

¹ Laboratório de Computação Científica Aplicada à Saúde Coletiva (LCCASC),
UNISANTOS, Santos, Brazil
henrique.gagliardi@gmail.com

² Programa de Mestrado em Saúde Coletiva, UNISANTOS, Santos, Brazil
quiron@unisantos.br

³ Programa de Mestrado em Informática, UNISANTOS, R. Carvalho de Mendonça, 144,
Santos, Brazil
fabricio@unisantos.br

Abstract. The main goal this paper is to describe a software simulating spatio-temporal Dengue epidemic spread based on the utilization of a generalized probabilistic cellular automata computational analysis as the dynamic model of spatial epidemiology. This epidemic spatial model permits to reproduce explicitly the interaction of two types of transmission mechanisms in terms of global and local variables, which in turn can be adjusted to simulate respectively the populational mobility and geographical neighborhood contacts. The resulting virtual laboratory was designed to run spatio-temporal simulation of the Dengue disease spreading based on local and global interactions among two distinct populations (humans and mosquitoes).

1 Introduction

Mathematical and statistical modeling, in particular, is valuable in the design and interpretation of epidemiological studies, but is not well suited to simulation of interventions and resulting changes of behaviour [1,2]. Extraordinary increase in speed of computer processors and availability of cheap memory now make it possible to provide better prediction and to define better strategies to overwhelm the spread of epidemics, giving rise to development of the new mathematical and computing methodologies. Such methodologies could serve as virtual “laboratories” for a new science of experimental epidemiology in which new population-level interventions could be designed, evaluated, and iteratively refined on simulated epidemics, with tangible benefits for real-world epidemic prevention and control efforts. Indeed, some developments have already appeared at these epidemiological/computational interfaces [3,4].

Particularly, the successful applications of Cellular Automata (CA) in the context of epidemic dynamic [2,5] have motivated us to develop a simulator based on the utilization of generalized probabilistic cellular automata [2], to study the spread of Dengue fever. The reason for this approach is that cellular automata have a significant role in epidemic modeling because each individual, or cell, or small region of space

"updates" itself independently (in parallel) allowing for the concurrent development of several epidemic spatial clusters, defining its new state based on the current state of its surrounding cells (locality) and on some shared laws of change [2,5].

Therefore, the main contribution of this paper is to present a software system that incorporates a general probabilistic cellular automata model [2] to describe the dynamics of Dengue transmission in a virtual urban environment. The basis of the transmission model is the simulation of stochastic inter-populations interactions (human and mosquitoes) including the infectious period in which individual interactions are straightforwardly described in terms of the local (physical contacts) and the global (populational mobility) modes of transmission [2]. At last, we shall discuss some simplifications of our model for Dengue spreading and to provide a modicum of validation for a specific location. We also compare simulation results with reports population data from dengue epidemic in real city. The software itself consists of four modules: the specification module, the simulation module, the visualization module and the analysis module. Particularly, the analysis module is used for spatio-temporal model analysis in the same environment, using a friendly user interface while at the same time providing considerable flexibility to analysis.

The remainder of this paper is organized as follows. In the next section we detail the model implemented in the software system. The simulator system itself is presented in section 3. Finally, conclusions and some directions for future work are stated in section 4.

2 The Epidemic Cellular Automata Model for Dengue

In this section, we will focus our discussion in the Dengue model, which is a human-vector iteration model. Consider a discrete dynamical system (discrete space and discrete time) where a population of N individuals is distributed on the sites of a lattice $M = m_{ij}$ — with i and j may vary from 1 to L ($N = L \times L$). Each individual site is assigned to receive three specific attributes: (1) a spatial address or lattice position (i,j) ; (2) a set of possible clinical status, specifying a clinic disease stage of each particular individual, which represent, for instance, the conditions of susceptible (subject to be infected by a contagious agent), infectious (effectively transmitter of contagious agents) and removed (recovered or immune); and finally (3) a period τ , specifying how many units of time an individual stays on a specific clinical status, i.e. how much time an infection disease can propagate through the contagious agent. The dynamic evolution of the population is described, step-by-step, by a set of interaction rules defined a priori, and assumes that each new configurational state of the system (described here by the lattice position (i,j) of each individual and by the instantaneous number of individuals in each particular state), depends only on its previous state.

The main assumption of our approach is that the Dengue transition rules are defined by two sets of states, each one representing the behavior of a distinct population. In the language of state-variable models, the humans are defined by a SEIR (Susceptible, Exposed, Infective and Recovered) model which when Exposed, the individual is infected but still not infective. The mosquito population is represented by a different SEI (Susceptible, Exposed and Infective) model as shown in Fig. 1.

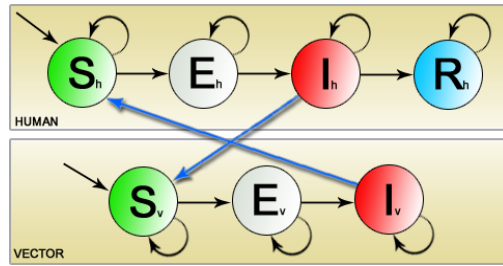


Fig. 1. The schematic model of Dengue spreading representing the stages of the disease for both populations, where *thick edges* represent the interaction among populations (mosquito bite) and the thin ones the internal transition of states in each model

The Dengue model has a singular characteristic, different from the other models based in cellular automata: the use of two overlapping interacting automata cellular grids to represent the human and the vector (*Aedes aegypti* mosquito) populations that transmit Dengue virus. Therefore, the neighborhood of each individual has a different meaning and is not defined around each individual cell in the same population but at an equivalent position at the overlapping one, i.e. a neighbor of a human cell is defined by the mosquitoes (vector) population and vice-versa. This special characteristic allows for the possibility of a human host being locally infected by a vector and to infect another mosquito. This interaction (a bite), however, never occurs directly among the neighbors of a same population because a human may only become infected by *Aedes aegypti* bite and a mosquito only becomes infected by biting an infective human. Moreover, here we propose a novel framework to model the spread of a Dengue outbreak. Besides defining internal state transition rules for each iterative population (human and mosquitoes), we define iterative rules between these two cellular automata (Fig. 2), reflecting them from one population to the other:

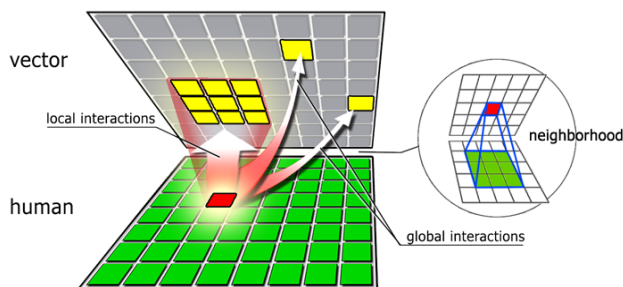


Fig. 2. The local and global influences are shown in this figure. The *pointed squares* represent the mosquitoes affected by the local and global human infective influence. The same kind of influences occurring in this bottom-up direction for human-vector interactions also occurs for the vector-humans in a top-down sense at each simulation update. We can see also in this figure a schematic representation of the *neighborhood* of a single element in a given automata network.

1. Any susceptible individual may become infected with probability p_S .
2. An infected individual becomes infective after an average latency time τ_E (assumed here as a constant, without loss of generality).
3. Infective individuals are removed deterministically from the system (becoming immune) after an infectious period $\tau_I > 0$, which is considered as a constant for all infected human individuals and infinity to the mosquito population.
4. Once in the removed class, the individual participate only passively in the spreading of the infection (eventually blocking the spreading process) by a period of immunity larger than the complete epidemic process.

The probability p_S of any susceptible individual (human or mosquito) to become infective is a superposition of: the local mode of transmission that incorporates the individual-based component from the perspective of the susceptible individuals, the actual (physical) contacts that each susceptible individual experiences; and the global influence due to intrinsic populational mobility, which may be viewed as resulting of a mean-field (discrete) approach, in the sense that the disease transmission to each susceptible individual also depends on the instantaneous total number of infectious individuals in the population. Therefore, one can assume that disease transmission occurs with a total infection probability p_S written as:

$$p_S = \Gamma p_G + \Lambda p_L, \tag{1}$$

where the pre-factors Γ and Λ are weight parameters tuning the short (cluster formation) and long-range (mean-field type) interactions; it is also required that $\Gamma + \Lambda = 1$ in order to satisfy the probabilistic requirement $0 \leq p_S \leq 1$. The global influence p_G represents the probability of a susceptible individual in a population (human or mosquito) to become infective due to the presence of $I(t)$ infected individuals in another population (mean-field) and it is related to the intrinsic mobility of the population elements on space, i.e. the ability of an element to establish contact with anyone of the another population and infect him. On the other hand, the local term $p_L = p_L(i, j)$ is the probability of a susceptible individual (located at the site (i,j)) being infected due to n infectives in the first and second neighborhoods (Moore neighborhood).

The resulting probabilities to each population are as follow:

Humans:

$$p_L = 1 - (1 - \lambda_m)^{n_m} \text{ and } p_G = \frac{\rho_h N_{mi}(t)}{N_m} \tag{2}$$

Mosquito:

$$p_L = 1 - (1 - \lambda_h)^{n_h} \text{ and } p_G = \frac{\rho_m N_{ih}(t)}{N_h} \tag{3}$$

Therefore, the system is governed by p_S (Equation 1) and τ , and its temporal evolution is determined by updating the lattice synchronously at each time step through the application of the three rules above. This model is available on a

simulator initially developed to help the comprehension and analysis of epidemics spread that we will be presenting in the following section.

3 Design and Implementation of the Dengue Epidemic Simulator

In this section, we discuss the design and implementation of an epidemic simulation tool, the technological choices we made and an overview of the general architecture of the simulation tool. We choose C++ as the programming language due to robustness and facilities offered as an object-oriented programming language and its possibility to append new models and tools to the system, which is upgraded in a modular way. To create grid animations during simulations we are using OpenGL.

The software system consists of four modules: the specification module, the simulation module, the visualization module and the analysis module. It is worth to note that the specification module generates the model specification which serves as input to both the simulation and visualization modules and the analysis module supports spatio-temporal model analysis.

A setup window enables the user to configure all the parameters of each available model. Thus, each model has its specific settings and many different kinds of scenarios can be simulated by adjusting its values. Through this specification module we can configure the period (in days) that the human will stay exposed and infected, the period that the vector stays incubated (exposed), the global and local probability of interaction inter grids, grids initial conditions (with one infected human/vector at the center or a percentage of individuals in different states of a full susceptible grid when the simulation starts for both populations) and the type of view we wish to use (perspective or orthogonal).

The simulation module is the responsible for execution of the model simulations. When executing a specified simulation, initially every element of the grids is set to the initial conditions defined in the specification module. Each element has an integer value that represents its state (0 (zero) for the susceptible, 1 to τ_E for the exposed (infected), $\tau_E + 1$ to $\tau_E + \tau_I$ for the infectives and $\tau_E + \tau_I + 1$ for the recovered). Therefore, each update is done by increasing the current time of the system and visiting each individual of both grids to update its states. If the individual current state is the susceptible state, it has a probability p_S to become infected (Equation 1), which is calculated by the sum of the global and local probabilities described by Equations 2 and 3, respectively. However, if its current state is not susceptible, its current integer value is increased (as schematized in Fig.1). After each updating, the visualization module displays an animation of the grids dynamics at the user's screen.

The visualization module shows a graphical animation of the spreading of the Dengue epidemics using the previously defined parameters (as shown in Fig. 3). It is possible to control and customize the stochastic realizations of the simulation process through navigator and velocity toolbars. It is also possible to change graphic windows settings or disable the animation to speed-up the simulation process.

Particularly, in the Fig. 3 we can see very interesting characteristics of Dengue model. In fact, we can see the mobility effects of a simulation with the same grid

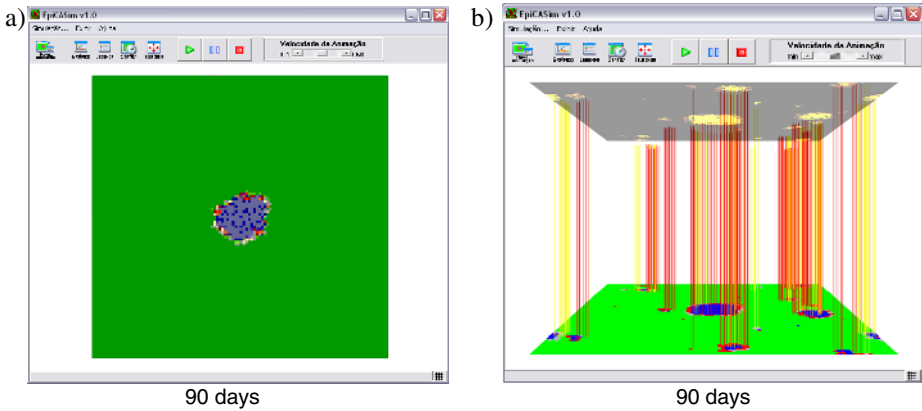


Fig. 3. Two Dengue cellular automata iterating in the perspective view. In (a) we see a simulation with the orthogonal view and in (b) the perspective view, where the *bottom grid* represents the human population while the *top grid* represents the vector population. The *connections* between the grids show the effective contact between different populations and it is also possible to see the global influence for Dengue spreading by the infective clusters geographically distributed in the grid.

parameters and λ values but with $\Gamma = 0$ for both networks (a system without global iterations) for (a) and $\Gamma = 0,2$ for the human population (a system with human global interactions with mosquitoes) for (b). Comparing both simulations we can see that the first requires a large amount of time to percolate the entire grid. In this case the spreading is geographically dependent. However, with just a small Γ value we can observe in (b) that the spreading is faster than (a) because the spreading becomes concurrent due the global moves among grids elements, what looks like a small world phenomena scenario [6] which elements relationships makes a spreading process faster by establishing shortcuts among distant elements (similar to the global influence). This space of parameters can be adequately explored through the analysis module (Fig. 4) by executing series of simulations varying the model parameters.

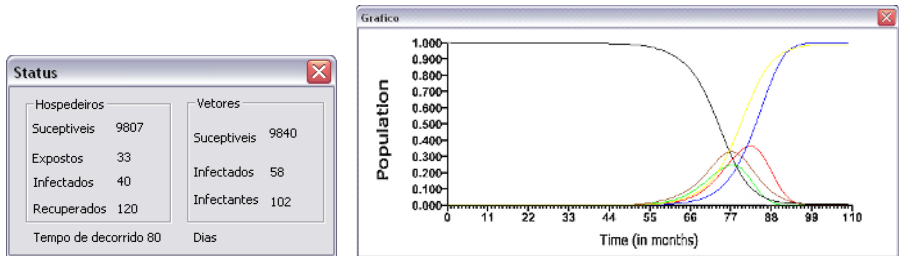


Fig. 4. The *Status* and the *Graphic* windows of the analysis module

In fact, a very interesting feature of the analysis module is the ability to see the progression of the epidemic spread and to generate series of stochastic realizations in which averages are calculated by the data generated in the simulated epidemic

process. It is then possible to visualize the average behavior of the system for a variable set of configurable intervals of parameters.

4 Conclusions and Future Work

This paper describes a simulator for the study of the spread of Dengue epidemic using a probabilistic cellular automata model. Strengths and weaknesses of software observed in practice show that there is a general need for improvement concerning the clarification of model structure, as well as the support for systematic validation and robustness tests. Furthermore, owing to its extreme simplicity, this formulation may be useful, in the sense of having the value of an approximation, to tackle problems in Dengue epidemiology.

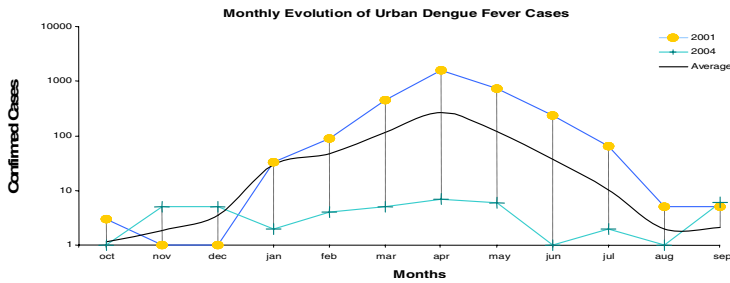


Fig. 5. The monthly evolution of confirmed cases in the years 2001 and 2004. Here we can observe the average behavior of urban Dengue fever in Ribeirão Preto city in the *dark line*.

As shown by the Dengue progress curves from Ribeirão Preto, a city in southeast Brazil (see Fig. 5) seasonal effects play a central role in disease spreading. In fact, the fastest Dengue spreading progress is observed from November through March (end of summer), a period associated to high temperatures and regular rains. The seasonal modulations are been included in the CA model through the variation in the motility of vectors as well as in the fraction of humans in the removed state. The remaining CA parameters were varied in order to compare the simulated and observed Dengue progress curves. We find in our virtual experiments that in the model space of parameters there are different value intervals in which the resulting qualitative behavior of the CA progress curves show the same functional behavior of the measured curves shown on Fig. 5. Even a surprisingly quantitative agreement between the CA and the observed Dengue progress curves were obtained in a restrict set of values of parameters. To be used as a decision-making tool, however, the simulator must be better parameterized and validated with actual epidemiological data.

Moreover, the possibility to simulate the evolution of a Dengue epidemic model over a digitalized city map is another desired feature that actually we are working on. This is an important issue, since it would be highly desirable to study and to visualize the infective individuals located in a layer over a city map updated along the simulation process.

At the present moment we are able to execute our simulations over a digital city map image where the Dengue model is constructed considering the urban regions (as in Fig. 6, here taken as a marginal region around the streets).

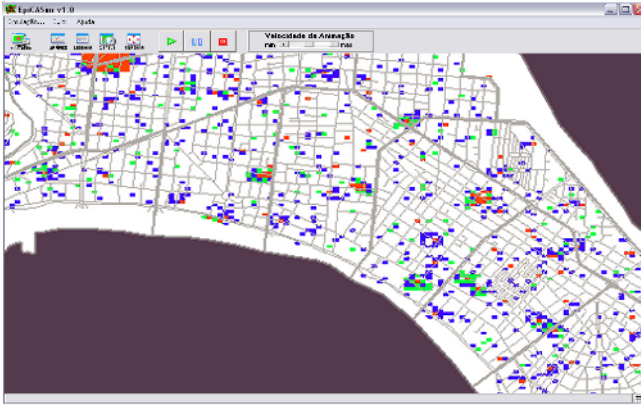


Fig. 6. A simulation of the Dengue model over Santos city map, a southeast costal Brazilian city which epidemic historical data will also be used in our future study

This is a useful feature to observe the geographical distribution of the simulation over a real city shaped grid. To the best of our knowledge, there are no similar results reported for this type of coupled cellular automata.

Acknowledgements

This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP): Proc. 02/03564-8.

References

1. Paterson, S., Viney, M. E.: The interface between epidemiology and population genetics, *Parasitology Today*, Vol.16. 12 (2000) 528.
2. Alves, D., Hass, V., Caliri, A.: The predictive power of R_0 in an epidemic probabilistic model. *Journal of Biological Physics*, Vol. 29. 1 (2003) 63.
3. Focks, D., A., et al. A simulation model of the epidemiology of urban dengue fever: Literature analysis, model development, preliminary validation, and samples of simulation results. *American Journal of Tropical Medicine and Hygiene*, Vol. 53. (1995) 489-506.
4. Stefano, B., Fukú, H., Lawniczak, A. T., Object-Oriented Implementation of CA/LGCA Modelling Applied To the Spread of Epidemics. *Canadian Conference on Electrical and Computer Engineering (IEEE)*, Vol.1. (2000) 26-31.
5. Boccaro, N., Cheong, K.: Automata Network SIR Models for Spread of Infectious Diseases in Populations of Moving Individuals. *J. Phys. A: Math and Gen.*, Vol.25. (2000), 2447.
6. Watts, D. J. and Strogatz, S. H., The Collective Dynamics of 'Small-World' Networks, *Nature*, Vol.393 (1998) 440.

A Picture for Complex Stochastic Boolean Systems: The Intrinsic Order Graph*

Luis González

University of Las Palmas de Gran Canaria,
Department of Mathematics, Research Institute IUSIANI,
35017 Las Palmas de Gran Canaria, Spain
luisglez@dma.ulpgc.es
<http://www.dma.ulpgc.es/profesores/personal/lgs/>

Abstract. Complex stochastic Boolean systems, depending on a large number n of statistically independent random Boolean variables, appear in many different scientific, technical or social areas. Each one of the 2^n binary states associated to such systems is denoted by its corresponding binary n -tuple of 0s and 1s, (u_1, \dots, u_n) , and it has a certain occurrence probability $\Pr\{(u_1, \dots, u_n)\}$. The ordering between the 2^n binary n -tuple probabilities, $\Pr\{(u_1, \dots, u_n)\}$, can be illustrated by a directed graph which “scales” them by decreasing order, the so-called intrinsic order graph. In this context, this paper provides a simple algorithm for iteratively drawing the intrinsic order graph, for any complex stochastic Boolean system and for any number n of independent random Boolean variables. The presentation is self-contained.

1 Introduction

This paper deals with the modelling of complex stochastic Boolean systems, that is, those complex systems which depend on a large number n of random Boolean variables x_1, \dots, x_n . These systems, with very simple components (the Boolean variables x_i only take two possible values: 0, 1) but complex overall behavior, can be found in many different knowledge areas: Wherever a stochastic Boolean phenomenon with n basic variables appears (Biology, Biochemistry, Climatology, Computer Science, Networks, Engineering, Economics, Sociology, etc.).

According to the usual terminology in Statistics, a stochastic Boolean system can be modeled by the n -dimensional Bernoulli distribution (see, e.g., [6]). This distribution consists on n random variables x_1, \dots, x_n , which only take two possible values, 0 or 1, with probabilities

$$\Pr\{x_i = 1\} = p_i, \quad \Pr\{x_i = 0\} = 1 - p_i \quad (1 \leq i \leq n),$$

so that the sample space is the set $\{0, 1\}^n$ of the 2^n binary n -tuples (binary strings), $u = (u_1, \dots, u_n)$, of 0s and 1s. Throughout this work, we assume that

* Partially supported by MEC (Spain) and FEDER. Grant contract: CGL2004-06171-C03-02/CLI.

the random variables x_i are mutually independent, so that the occurrence probability of each binary n -tuple can be easily computed as the product

$$\Pr \{(u_1, \dots, u_n)\} = \prod_{i=1}^n \Pr \{x_i = u_i\} = \prod_{i=1}^n p_i^{u_i} (1 - p_i)^{1-u_i}, \tag{1}$$

that is, $\Pr \{(u_1, \dots, u_n)\}$ is the product of factors p_i if $u_i = 1$, $1-p_i$ if $u_i = 0$.

The following natural question immediately arises in the study of stochastic Boolean systems: How can we order the 2^n binary strings (u_1, \dots, u_n) by decreasing/increasing order of their occurrence probabilities? Obviously, this question has a relevant, theoretical and practical, interest. However, in spite of the simplicity of Equation (1) to compute the probabilities $\Pr \{(u_1, \dots, u_n)\}$, this is not a simple question due to its exponential nature. To avoid this obstacle, in [3, 4] the authors have established a simple positional criterion, the so-called *Intrinsic Order Criterion* (IOC), that *a priori* assures us that for certain pairs of binary n -tuples $u, v \in \{0, 1\}^n$ (exactly for those pairs whose bits satisfy IOC), the inequality $\Pr \{u\} \geq \Pr \{v\}$ *intrinsically* holds, that is, $\Pr \{u\} \geq \Pr \{v\}$ for any values of the parameters p_i satisfying certain non restrictive hypothesis. Moreover, in [1, 2] the author has “roughly” constructed the intrinsic order graph for the first values of n , just by direct application of IOC. However, a general method for easily constructing this graph for all $n \in \mathbb{N}$, has not yet been provided.

In this context, *the main goal of this paper is to provide a simple algorithm for recursively drawing the intrinsic order graph, for any number n of Boolean variables. This directed graph is a very useful picture for all complex stochastic Boolean systems (related to any knowledge area), because it “scales” the binary n -tuples of 0s and 1s, $(u_1, \dots, u_n) \in \{0, 1\}^n$, by decreasing order of their occurrence probabilities.* In Sect. 2, we describe all previous results required for making this paper self-contained. The rest of the paper is devoted to our new approach. In Sect. 3, we present a simple characterization of the covering relation of the intrinsic order. Finally, in Sect. 4, from this characterization we derive the algorithm for drawing the intrinsic order graph.

2 The Intrinsic Order

In [3, 4], we have established the following characterization theorem that allows us to compare two given binary string probabilities, $\Pr \{u\}$ and $\Pr \{v\}$, without computing them.

Theorem 1 (The intrinsic order theorem). *Let x_1, \dots, x_n be n independent Bernoulli variables, with parameters $p_i = \Pr \{x_i = 1\}$ ($1 \leq i \leq n$) satisfying:*

$$0 < p_1 \leq \dots \leq p_n \leq \frac{1}{2}. \tag{2}$$

Then, the probability of the n -tuple $(u_1, \dots, u_n) \in \{0, 1\}^n$ is intrinsically greater than or equal to the probability of the n -tuple $(v_1, \dots, v_n) \in \{0, 1\}^n$ (that is, for

all set of parameters $\{p_i\}_{i=1}^n$ such that (2)) if, and only if, the matrix

$$M_v^u := \begin{pmatrix} u_1 & \cdots & u_n \\ v_1 & \cdots & v_n \end{pmatrix}$$

either has no $\binom{1}{0}$ columns, or for each $\binom{1}{0}$ column there exists (at least) one corresponding preceding $\binom{0}{1}$ column (IOC).

Remark 1. In the following, we assume that the parameters p_i always satisfy condition (2). Note that this hypothesis is not restrictive for practical applications because, if for some $i : p_i > \frac{1}{2}$, then we only need to consider the variable $\bar{x}_i = 1 - x_i$, instead of x_i . Next, we order the n Bernoulli variables by increasing order of their probabilities.

Remark 2. The $\binom{0}{1}$ column preceding to each $\binom{1}{0}$ column is not required to be necessarily placed at the immediately previous position, but just at previous position.

Remark 3. The term *corresponding*, used in Theorem 1, has the following meaning: For each two $\binom{1}{0}$ columns in matrix M_v^u , there must exist (at least) two *different* $\binom{0}{1}$ columns preceding to each other. In other words, IOC requires that for each $\binom{1}{0}$ column C in matrix M_v^u , if we denote by $n_1^0(C)$ ($n_0^1(C)$, resp.) the number of $\binom{0}{1}$ columns ($\binom{1}{0}$ columns, resp.) preceding C , then $n_1^0(C) > n_0^1(C)$.

Theorem 1 naturally leads to the following partial order relation on the set $\{0, 1\}^n$. The so-called intrinsic order (because it only depends of the relative positions of 0s and 1s in the binary n -tuples u, v , but not on the basic probabilities p_i), will be denoted by “ \preceq ”, and we shall denote by P_n the partially ordered set (poset, for short) $(\{0, 1\}^n, \preceq)$. See [5] for more details about posets.

Definition 1. For all $u, v \in \{0, 1\}^n$

$u \succeq v$ iff $\Pr\{u\} \geq \Pr\{v\}$ for all set $\{p_i\}_{i=1}^n$ s.t. (2) iff M_v^u satisfies IOC.

Example 1. $(0, 0, 1, 1) \succeq (1, 1, 0, 0)$ because matrix

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

satisfies IOC (Remark 2).

Example 2. $(0, 1, 1, 0) \not\succeq (1, 0, 0, 0)$ and $(1, 0, 0, 0) \not\succeq (0, 1, 1, 0)$ because

$$\text{neither } \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \text{ nor } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

satisfies IOC (Remark 3).

From now on, we shall indistinctly denote any n -tuple $u \in \{0, 1\}^n$ by its binary representation (u_1, \dots, u_n) or by its decimal representation $u_{(10)}$. Also, we shall denote by “ \leq_{lex} ” the usual lexicographic order (truth-table order) between the binary n -tuples, with the convention that $0 < 1$, i.e.,

$$(u_1, \dots, u_n) \equiv u_{(10)} := \sum_{i=1}^n 2^{n-i} u_i, \quad u \leq_{lex} v \Leftrightarrow u_{(10)} \leq v_{(10)}.$$

The next corollary (see [2] for the proof) states the relation between the (partial) intrinsic order “ \preceq ” and the (total) lexicographic order “ \leq_{lex} ”.

Corollary 1. *For all $u, v \in \{0, 1\}^n$*

$$u \succeq v \Rightarrow u \leq_{lex} v, \text{ i.e., } u \succeq v \Rightarrow u_{(10} \leq v_{(10}.$$

3 Covering Relation in P_n

As is well-known [5], every finite poset is completely determined by its cover relations. Specifically, with respect to our poset P_n , we say that v is covered by u (or u covers v), denoted by $v \triangleleft u$ (or $u \triangleright v$) if $v \prec u$ and there is no $w \in \{0, 1\}^n$ such that $v \prec w \prec u$. For instance, in $P_2 = (\{0, 1\}^2, \preceq)$, we have $(1, 1) \triangleleft (1, 0) \triangleleft (0, 1) \triangleleft (0, 0)$ because $(1, 1) \prec (1, 0) \prec (0, 1) \prec (0, 0)$ (Definition 1, Theorem 1) with no other elements between them. However, $(1, 1) \prec (0, 1)$ but $(1, 1) \not\triangleleft (0, 1)$, because $(1, 0)$ is between them. In this section, we provide a simple matrix characterization of the covering relation associated to the intrinsic order. This condition for “ \triangleleft ”, required for drawing the intrinsic order graph in the next section, will be obviously a particular case of (more restrictive than) the IOC condition for “ \preceq ”. First, we need the following lemma.

Lemma 1. *Let $u, v \in \{0, 1\}^n$ and $1 \leq i \leq n - 1$.*

- (i) *If $v \preceq u$ then $(v_1, \dots, v_i) \preceq (u_1, \dots, u_i)$.*
- (ii) *If $v \preceq u$ and $\begin{pmatrix} u_1 \dots u_i \\ v_1 \dots v_i \end{pmatrix}$ has the same number of $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ columns as $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ columns then $(v_{i+1}, \dots, v_n) \preceq (u_{i+1}, \dots, u_n)$.*
- (iii) *If $v \preceq u$ and the columns $\begin{pmatrix} u_i \\ v_i \end{pmatrix}$ and $\begin{pmatrix} u_{i+1} \\ v_{i+1} \end{pmatrix}$ are both different from $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then the matrix obtained by permuting these two columns of M_v^u , keeping its other columns, satisfies IOC.*
- (iv) *If $v \preceq u$, $\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\begin{pmatrix} u_j \\ v_j \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ for some $j > i$, and there is no k such that $i < k < j$ and $\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then the matrix obtained by replacing the i -th and j -th columns of M_v^u by $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ or $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, keeping its other columns, satisfies IOC.*

Proof. Using the matrix description IOC of the intrinsic order (Theorem 1), the proof is straightforward. □

Theorem 2 (Covering relation in P_n). *Let $n \geq 1$ and $u, v \in \{0, 1\}^n$. Then $v \triangleleft u$ if and only if either*

$$M_v^u = \begin{pmatrix} u_1 \dots u_{n-1} & 0 \\ u_1 \dots u_{n-1} & 1 \end{pmatrix}, \text{ or} \tag{3}$$

$$M_v^u = \begin{pmatrix} u_1 \dots u_{i-1} & 0 & 1 & u_{i+2} \dots u_n \\ u_1 \dots u_{i-1} & 1 & 0 & u_{i+2} \dots u_n \end{pmatrix} \quad (1 \leq i \leq n - 1), \tag{4}$$

where we assume that in case (3) the $n - 1$ first columns of matrix M_v^u are deleted if $n = 1$; while in case (4) the $i - 1$ first (the $n - i - 1$ last, resp.) columns of matrix M_v^u are deleted if $i = 1$ (if $i = n - 1$, resp.).

Proof. Sufficient condition. First, note that, for both cases (3) and (4), $v \prec u$ since the two corresponding matrices M_v^u obviously satisfy IOC. So, for both cases, we only need to prove that if $v \preceq w \preceq u$ then $w = u$ or $w = v$.

If (3) holds then using Corollary 1, we have $v \preceq w \preceq u \Rightarrow u \leq_{lex} w \leq_{lex} v$ and then $w = u$ or $w = v$.

If (4) holds then using Corollary 1, we have $v \preceq w \preceq u \Rightarrow u \leq_{lex} w \leq_{lex} v$ and then two subcases are possible:

$$(w_1, \dots, w_{i+1}) = \begin{cases} (u_1, \dots, u_{i-1}, 0, 1) & \text{(4)-a,} \\ (u_1, \dots, u_{i-1}, 1, 0) & \text{(4)-b.} \end{cases}$$

- If $i = n - 1$, then (4)-a is equivalent to say $w = u$ while (4)-b is equivalent to say $w = v$. Then, $w = u$ or $w = v$.
- Otherwise, if $1 \leq i < n - 1$, then using Lemma 1-(ii), we have: On one hand, since $v \preceq w$ and matrix $\begin{pmatrix} w_1 & \dots & w_{i-1} & w_i & w_{i+1} \\ u_1 & \dots & u_{i-1} & 1 & 0 \end{pmatrix}$ has the same number of $\binom{1}{0}$ columns as $\binom{0}{1}$ columns (namely, 1 if (4)-a holds and 0 if (4)-b holds) then we get $(u_{i+2}, \dots, u_n) \preceq (w_{i+2}, \dots, w_n)$. On the other hand, since $w \preceq u$ and matrix $\begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 \\ w_1 & \dots & w_{i-1} & w_i & w_{i+1} \end{pmatrix}$ has the same number of $\binom{1}{0}$ columns as $\binom{0}{1}$ columns (namely, 0 if (4)-a holds and 1 if (4)-b holds) then we get $(w_{i+2}, \dots, w_n) \preceq (u_{i+2}, \dots, u_n)$. So, due to the antisymmetry of the intrinsic order we obtain $(w_{i+2}, \dots, w_n) = (u_{i+2}, \dots, u_n)$, and then $w = u$ (if (4)-a holds) or $w = v$ (if (4)-b holds).

Necessary condition. We provide a constructive proof by finding $w \in \{0, 1\}^n$ yielding $v \not\preceq w \not\preceq u$, for all $v \prec u$ such that the pattern of matrix M_v^u is different from (3) and (4). First, note that since $v \prec u$ we assure (Corollary 1) that $u <_{lex} v$, i.e., the first column of M_v^u different from $\binom{0}{0}$ and $\binom{1}{1}$ is $\binom{0}{1}$. Let $\binom{u_i}{v_i}$ be this column. Then

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & u_{i+1} & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 1 & v_{i+1} & v_{i+2} & \dots & v_n \end{pmatrix}, \tag{5}$$

where we can assure that $1 \leq i < n$, because if $i = n$ then M_v^u in (5) would be like (3), and we assume that the $i - 1$ first (the $n - i - 1$ last, resp.) columns of matrix (5) are deleted if $i = 1$ (if $i = n - 1$, resp.). The same assumption is established for the $i - 1$ first (the $n - i - 1$ last, resp.) components of the chosen vectors w . We distinguish the following four cases.

(1) $\binom{u_{i+1}}{v_{i+1}} = \binom{0}{0}$. In this case we choose

$$w = (u_1, \dots, u_{i-1}, 0, 1, v_{i+2}, \dots, v_n), \text{ so that } v \neq w \neq u \text{ and}$$

$$M_v^w = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \\ u_1 & \dots & u_{i-1} & 1 & 0 & v_{i+2} & \dots & v_n \end{pmatrix} \text{ and } M_w^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 0 & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \end{pmatrix}$$

satisfy IOC: the first one obviously; the second one due to Lemma 1-(iii).

(2) $\binom{u_{i+1}}{v_{i+1}} = \binom{1}{1}$. In this case we choose

$$w = (u_1, \dots, u_{i-1}, 1, 0, u_{i+2}, \dots, u_n), \text{ so that } v \neq w \neq u \text{ and}$$

$$M_v^w = \begin{pmatrix} u_1 & \dots & u_{i-1} & 1 & 0 & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 1 & 1 & v_{i+2} & \dots & v_n \end{pmatrix} \text{ and } M_w^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 1 & 0 & u_{i+2} & \dots & u_n \end{pmatrix}$$

satisfy IOC: the first one due to Lemma 1-(iii); the second one obviously.

(3) $\binom{u_{i+1}}{v_{i+1}} = \binom{0}{1}$. Here we distinguish two subcases:

– (3.1) M_v^u has no $\binom{1}{0}$ columns. In this subcase we choose

$$w = (u_1, \dots, u_{i-1}, 0, 1, v_{i+2}, \dots, v_n), \text{ so that } v \neq w \neq u \text{ and}$$

$$M_v^w = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \\ u_1 & \dots & u_{i-1} & 1 & 1 & v_{i+2} & \dots & v_n \end{pmatrix} \text{ and } M_w^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 0 & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \end{pmatrix}$$

satisfy IOC, since they have no $\binom{1}{0}$ columns.

– (3.2) M_v^u has at least one $\binom{1}{0}$ column. Let $\binom{u_j}{v_j}$, $i + 2 \leq j \leq n$, be the left-most $\binom{1}{0}$ column of M_v^u . In this subcase we choose

$$w = (u_1, \dots, u_{i-1}, 0, 1, v_{i+2}, \dots, v_{j-1}, 1, v_{j+1}, \dots, v_n), \text{ so that } v \neq w \neq u$$

and with the assumption that the substring $(v_{i+2}, \dots, v_{j-1})$ of w is deleted if $j = i + 2$. Then

$$M_v^w = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_{j-1} & 1 & v_{j+1} & \dots & v_n \\ u_1 & \dots & u_{i-1} & 1 & 1 & v_{i+2} & \dots & v_{j-1} & 0 & v_{j+1} & \dots & v_n \end{pmatrix},$$

$$M_w^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 0 & u_{i+2} & \dots & u_{j-1} & 1 & u_{j+1} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_{j-1} & 1 & v_{j+1} & \dots & v_n \end{pmatrix}$$

satisfy IOC: the first one obviously; the second one due to Lemma 1-(iv).

(4) $\binom{u_{i+1}}{v_{i+1}} = \binom{1}{0}$. Recall that for all possible cases (1)-(4) we assured that $1 \leq i < n$. Moreover, in this fourth case we can assure that $1 \leq i < n - 1$ and $(v_{i+2}, \dots, v_n) \neq (u_{i+2}, \dots, u_n)$, because if $i = n - 1$ or if $i < n - 1$ with $(v_{i+2}, \dots, v_n) = (u_{i+2}, \dots, u_n)$ then, in both cases, M_v^u in (5) would have the pattern (4). In this last case we choose

$$w = (u_1, \dots, u_{i-1}, 0, 1, v_{i+2}, \dots, v_n), \text{ so that } v \neq w \neq u \text{ and}$$

$$M_v^w = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \\ u_1 & \dots & u_{i-1} & 1 & 0 & v_{i+2} & \dots & v_n \end{pmatrix} \text{ and } M_w^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & 1 & u_{i+2} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 0 & 1 & v_{i+2} & \dots & v_n \end{pmatrix}$$

satisfy IOC: the first one obviously; the second one due to Lemma 1-(iv). □

4 The Intrinsic Order Graph

The classical picture of a poset P is its Hasse diagram. This is a directed graph (digraph, for short) whose vertices are the elements of P , whose edges are the cover relations, and with the usual convention that if $v \prec u$ then u is drawn

“above” v [5]. For instance, the digraph of P_1 is $\begin{matrix} 0 \\ | \\ 1 \end{matrix}$, since $1 \triangleleft 0$. The next theorem provides us with a fast algorithm for iteratively building up the Hasse diagram of P_n , for all $n > 1$, from the Hasse diagram of P_1 . The binary n -tuples are represented by their decimal numbering.

Theorem 3 (Iterative construction of P_n from P_1). *For all $n > 1$, the digraph of $P_n = \{0, \dots, 2^n - 1\}$ can be drawn simply by adding to the digraph of $P_{n-1} = \{0, \dots, 2^{n-1} - 1\}$ its isomorphic copy $2^{n-1} + P_{n-1} = \{2^{n-1}, \dots, 2^n - 1\}$. This addition must be performed placing the powers of 2 at consecutive levels of the Hasse diagram of P_n . Finally, the edges ($u \triangleright v$) connecting one vertex u of P_{n-1} with one vertex v of $2^{n-1} + P_{n-1}$ is given by the set of vertex pairs*

$$\{(u, v) / u \triangleright v\} = \{(u_{(10)}, 2^{n-2} + u_{(10)}) / 2^{n-2} \leq u_{(10)} \leq 2^{n-1} - 1\}.$$

Proof. For all $n > 1$, consider the following partition of $\{0, 1\}^n$

$$\begin{aligned} \{0, 1\}^n &= (\{0\} \times \{0, 1\}^{n-1}) \cup (\{1\} \times \{0, 1\}^{n-1}), \text{ i.e.,} \\ \{0, \dots, 2^n - 1\} &= \{0, \dots, 2^{n-1} - 1\} \cup \{2^{n-1}, \dots, 2^n - 1\}. \end{aligned}$$

On one hand, note that according to IOC (Theorem 1), the subposets of P_n , $(\{0\} \times \{0, 1\}^{n-1}, \preceq)$ and $(\{1\} \times \{0, 1\}^{n-1}, \preceq)$, are both isomorphic to P_{n-1} in the following sense. For all $(u_1, \dots, u_{n-1}), (v_1, \dots, v_{n-1}) \in \{0, 1\}^{n-1}$

$$\begin{aligned} (u_1, \dots, u_{n-1}) \preceq (v_1, \dots, v_{n-1}) &\Leftrightarrow (0, u_1, \dots, u_{n-1}) \preceq (0, v_1, \dots, v_{n-1}) \\ &\Leftrightarrow (1, u_1, \dots, u_{n-1}) \preceq (1, v_1, \dots, v_{n-1}). \end{aligned} \tag{6}$$

On the other hand, note that according to Theorem 2, we have for all $n > 1$

$$(0, \dots, 0, 1) \triangleright (0, \dots, 1, 0) \triangleright \dots \triangleright (1, 0, \dots, 0), \text{ i.e., } 2^0 \triangleright 2^1 \triangleright \dots \triangleright 2^{n-1}. \tag{7}$$

Equations (6) and (7) prove that P_n can be iteratively built up by adding to the digraph of P_{n-1} its isomorphic copy $2^{n-1} + P_{n-1}$, placing the powers of 2 at consecutive levels of the Hasse diagram of P_n . Finally, we prove the last assertion about the edges connecting the two mentioned subposets of P_n . Let $u \in \{0\} \times \{0, 1\}^{n-1} \equiv P_{n-1}$, $v \in \{1\} \times \{0, 1\}^{n-1} \equiv 2^{n-1} + P_{n-1}$ and put $u = (0, u_1, \dots, u_{n-1})$, $v = (1, v_1, \dots, v_{n-1})$. Using Theorem 2 and taking into account that the pattern (3) is not possible for matrix M_v^u because $n \geq 2$, we have

$$\begin{aligned} u \triangleright v &\Leftrightarrow M_v^u = \begin{pmatrix} 0 & u_1 & \dots & u_{n-1} \\ 1 & v_1 & \dots & v_{n-1} \end{pmatrix} \text{ has the pattern (4)} \\ &\Leftrightarrow \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and, if } n > 2 \text{ then } v_i = u_i \text{ for all } i = 2, \dots, n - 1 \\ &\Leftrightarrow u = \begin{cases} (0, 1) & \text{if } n = 2 \\ (0, 1, u_2, \dots, u_{n-1}) & \text{if } n > 2 \end{cases}, v = \begin{cases} (1, 0) & \text{if } n = 2 \\ (1, 0, u_2, \dots, u_{n-1}) & \text{if } n > 2 \end{cases} \end{aligned}$$

$$\Leftrightarrow (0, 1, 0, \dots, 0)_{(10)} \equiv 2^{n-2} \leq u_{(10)} \leq 2^{n-1} - 1 \equiv (0, 1, 1, \dots, 1)_{(10)}$$

and

$$v_{(10)} = u_{(10)} - 2^{n-2} + 2^{n-1} = u_{(10)} + 2^{n-2}$$

and this concludes the proof. □

Fig. 1 illustrates the iterative construction of the digraph of P_n ($n = 1, 2, 3, 4$) from P_1 , denoting all the binary n -tuples by their decimal numbering.

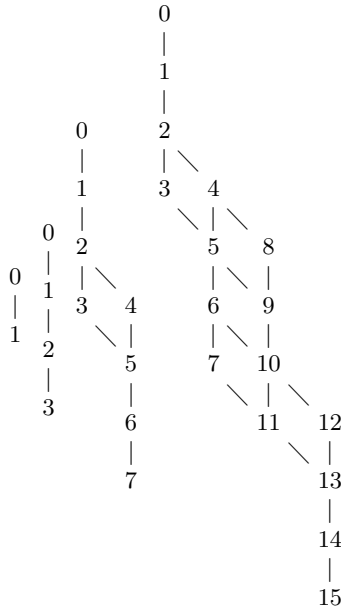


Fig. 1. The intrinsic order graph for $n = 1, 2, 3, 4$

References

1. González, L.: A new method for ordering binary states probabilities in Reliability and Risk Analysis. *Lect. Notes Comput. Sc.* **2329**(1) (2002) 137-146
2. González, L.: N-tuples of 0s and 1s: Necessary and sufficient conditions for intrinsic order. *Lect. Notes Comput. Sc.* **2667**(1) (2003) 937-946
3. González, L., Galván, B., García, D.: Sobre el análisis computacional de funciones Booleanas estocásticas de muchas variables. In: González, L., Sendra, J.R. (eds.): *Proc. Primer Encuentro de Álgebra Computacional y Aplicaciones (EACA-95)*. Santander (1995) 45-55
4. González, L., García, D., Galván, B.J.: An intrinsic order criterion to evaluate large, complex fault trees. *IEEE Trans. Reliability* **53**(3) (2004) 297-305
5. Stanley, R.P.: *Enumerative Combinatorics. Volume 1*. Cambridge University Press, Cambridge (1997)
6. Stuart, A., Ord, J.K.: *Kendall's Advanced Theory of Statistics. Volume 1: Distribution Theory*. Oxford University Press, New York (1998)

Evolutionary Spatial Games Under Stress

J. Alonso¹, A. Fernández¹, and H. Fort²

¹ Universidad de la República, Institute of Physics, Facultad de Ingeniería, Julio Herrera y Reissig 565, 11300 Montevideo, Uruguay

² Universidad de la República, Institute of Physics, Facultad de Ciencias, Iguá 4225, 11400 Montevideo, Uruguay

Abstract. We analyse different evolutionary spatial games, in which the pressure of the environment is taken into account, using binary cellular automata. The agents are unconditional players: at each time step a given cell cooperates (play C) or defects (play D) against all its neighbours. The pressure of the environment is implemented by requiring a minimum score U_{min} , representing indispensable resources (nutrients, energy, revenues, etc.) for an individual to prosper. Therefore a cell, instead of evolving just by adopting the state of its most successful neighbour, also takes into account if the "winner" gets a score above or below U_{min} . In the latter case it has a probability of adopting the opposite state. Besides the paradigmatic and widely used Prisoner's Dilemma (PD), two other games are analysed: the Hawk-Dove (H-D), popular in biology, and the Stag Hunt (SH) that recently came into favour in social sciences. The effect of the environmental stress is particularly dramatic in the case of the PD: it allows the evolution of cooperation for payoff matrices where defection was the rule for simple unconditional strategy players. Finally, we discuss a more sophisticated model version in which the ordinary evolutionary recipe of copying the most successful neighbour is supplemented with a "win-stay, lose-shift" criterion. This model variant, for a restricted region of the parameter space, produces critical scaling laws.

1 Introduction

Cooperation and competition in nature represent two faces of a same coin. Progress seems to need both: competition promotes efficiency at the individual level but cooperation is indispensable to achieve collective goals.

Game theory is a branch of mathematics concerned with the analysis of conflict situations. *Social dilemma* games demonstrated to be a suitable approach to describe this tension between individual and global interests in social sciences [1] as well in biology [2].

In social dilemma games each individual ("player") receives a higher payoff for a socially defecting choice than for a socially cooperative choice no matter what the other individuals in society do. But, at the same time, all individuals would be better off if all cooperate than if all defect. The most popular of such games is the *Prisoner's Dilemma* (PD) [3]. Imagine two players, each confronting two choices: cooperate (C) or defect (D), and each makes his choice without

knowing what the opponent will do. The possible outcomes are: 1) they can both cooperate: (C,C) and get the "reward" for mutual cooperation R , 2) they can both defect: (D,D) and get the "punishment" P for mutual defection or 3) one of them cooperates and the other defects: (C,D); in that case the one who played C gets the "sucker's payoff" S while his opponent gets the "temptation to defect" T . Hence we have a 2×2 *payoff matrix* with the payoffs obeying the chain of inequalities $T > R > P > S$. So independently of what the other player does, defection D yields a higher payoff than cooperation C: if your opponent defects, and you cooperate you will end up with the worst payoff. On the other hand, even if your opponent cooperates, you should defect because in that case your payoff is T which is higher than R . Nevertheless, playing both D they get P which is worst than R .

A more interesting situation arises when the game is played repeatedly. In this iterated Prisoner's Dilemma (IPD), if players have memory (at least from the immediately previous contest), there are several strategies that outperform the dominant one-shot strategy [D,D] and lead to some non-null degree of cooperation. For instance, in the strategic tournaments organised by Axelrod [1], [4] in the 80s, the winner strategy was 'TIT FOR TAT' (TFT), which plays C on the first move, and on all subsequent moves copy the choice of its opponent on the previous move. Afterwards a different computer tournament was carried out by Nowak and Sigmund [6]. In that case the winner strategy was the one previously named *simpleton* by Rapoport and Chamah [7] or *Pavlov* by D. and V. Kraines [8], because if its action result in a high payoff (T or R) it keeps it, but otherwise changes it.

On the other hand, territoriality by itself provides an alternative way to escape from iterated evolutionary game dilemmas without resorting to memory of past encounters conditioning the behaviour, which are *conditional strategies* (like TFT, Pavlov, etc.). This was shown by Nowak and May [9] who proposed a cellular automaton in which agents are binary *unconditional strategists*, that is: at each time step a given cell play versus all of their neighbours either C or D. In the next generation, a given cell adopts the state of the most successful cell of the neighbourhood (the one that collected the highest score among the cell itself and its neighbours). The game they considered is a simplified version of the PD in which the punishment P and the sucker's payoff S are equal ¹, implying then a "weak dilemma" (maximum punishment for mutual defection). Specifically they fixed $R=1$ and $P=S=0$ and found that the fraction of cooperators c varies from 1 to 0 as the temptation increases from $T=1$ to $T=2$. However, a problem with this simple model is that c drops to zero for ordinary non weak dilemmas in which all the individuals end playing D.

In the present work we propose a simple extension of the unconditional binary C.A. that allows the evolution of cooperation not just for weak dilemmas but for ordinary ones, where P and S can be quite different. The basic idea is that individuals need to collect, when playing with their z neighbours, a score

¹ Indeed this game is the frontier between the PD and the game known as *chicken* (see below).

above certain threshold U_{min} in order to prosper. In an ecosystem U_{min} represents the minimal resources (nutrients, energy, etc.) for surviving, alternatively, in markets it may represent the minimum revenues that make investments profitable. Provided the difference between P and S is sufficiently large, defecting individuals are on average the most successful in the first rounds. Nevertheless, if $U_{min} > zP$, they perform very poorly when surrounded by an entire neighbourhood of D's (something that in the real world this may be equivalent to death). Therefore we include a probability p of adopting the state opposite to the one of the most successful neighbour when its score is below U_{min} .

In order to minimise the number of model parameters we use a two-parameter payoff matrix and we fix the worst payoff $X^{(4)}$ to 0 and the second best $X^{(2)}$ to 1 (for example $S = 0$ and $R = 1$ for the PD). We explore a subspace of the space of parameters $\{T, P, U_{min}, p\}$ characterising the PD and measure the asymptotic (after a transient) fraction of cooperators. Furthermore, changing the rank order of the 4 payoffs we explore two other dilemma games. First, when the damage from mutual defection in the PD is increased so that it finally exceeds the damage suffered by being exploited, *i.e.* $T > R > S > P$, the new game is called the *chicken* or *Hawk-Dove* (H-D) game [2]. The 'Hawk' and 'Dove' allude to the two alternative behaviours displayed by animals in confrontation: hawks are expected to fight for a resource and will injure or kill their opponents (*i.e.* play D), doves, on the other hand, only bluff and do not engage in fights to the death (*i.e.* play C). This game applies thus to situations such that mutual defection (Hawk vs. Hawk) is the worst possible outcome for both as it happens in most of animal contests. Second, a nowadays popular game in social sciences is the *Stag Hunt* [10], corresponding to the payoffs rank order $R > T > P > S$ *i.e.* when the reward for mutual cooperation surpasses the temptation to defect. The name of the game derives from a metaphor invented by the French philosopher Jean Jacques Rousseau: Two hunters can either jointly hunt a stag or individually hunt a rabbit. Hunting stags is quite challenging and requires mutual cooperation. If either of them hunts a stag alone, the chance of success is minimal. Hunting stags is most beneficial for society but requires trust among its members.

2 The Model

The individuals (people, firms, bacteria, etc.) are represented by cells of a two-dimensional binary cellular automaton (CA). That is, each cell can take two possible values 0 or 1 corresponding to unconditional strategies for playing versus their neighbours: cooperate (C) or defect (D).

In this work we restrict ourselves to a) the *von Neumann neighbourhood* ($z = 4$ neighbouring cells: the cell above and below, right and left from a given cell) and b) the *Moore neighbourhood* ($z = 8$ neighbouring cells surrounding a given cell). Typical grid sizes range from 50×50 to 500×500 . Periodic boundary conditions are used. The score of a given player is the sum of all the payoffs he gets against each neighbour. The dynamic is synchronous: all the agents update their states simultaneously at the end of each CA-step.

In the CA of Ref. [9] natural selection is implemented very simply: each player adopts the strategy of the most successful neighbour (who got U^{msn}). Here, we consider the following modified evolutionary rule: If $U^{msn} > U_{min}$, then the player adopts the strategy of the most successful neighbour in the next generation. Otherwise, the player has a small probability p of adopting the opposite strategy. The rationale for this is that copying the most successful neighbour, when its payoff doesn't reach a critical threshold, may not be the most efficient strategy from an evolutionary point of view.

3 Results

After a transient the system reaches a steady state with a definite value c for the fraction of agents playing C. The duration of the transient depends on the lattice size and the neighbourhood. For instance for a 50×50 lattice and $z = 8$ it last typically between 100 and 200 rounds.

To avoid dependence on the initial conditions, the measures correspond to averages over an ensemble of 100 systems ² with arbitrary conditions.

Here, we present results for a subspace of the parameter space: the third best payoff is fixed to $X^{(3)}=0.5$. This value is chosen such that in the case of the PD, a punishment of $P=0.5$ implies, without any doubt, a non weak dilemma ³. We studied several values of the probability parameter and the ones reported here are always for $p = 0.1$. The best payoff $X^{(1)}$ is varied between 1 and 2. Finally, since $U_{min} < zX^{(3)}$ has no effects and $U_{min} > zX^{(1)}$ doesn't make sense (since no one can reach this threshold), the parameter space reduces to the square plane $X^{(1)} - U_{min}$ delimited by $1 \leq X^{(1)} \leq 2$ and $zX^{(3)} \leq U_{min} \leq zX^{(1)}$.

3.1 Prisoner's Dilemma: $S=0$, $P=0.5$, $R=1$ and $1 < T \leq 2$

Figs. 1.a and 1.b show c as a function of T and U_{min} . Basically three regions can be distinguished in the plots: i) A *stepladder region* emerges from the right border $U_{min} = zP$. ii) For not too large values of T and U_{min} there is a high *peak* of cooperation, delimited at the left by $U_{min} = zR=z$ (when all the cells play C). iii) Finally, beyond $U_{min}=zR=z$ c reaches a plateau delimited by the straight line $U_{min}(T)=zT$.

To understand the 3 different regions it is convenient to consider a small deviation from the minimum T : $T = 1 + \epsilon$. Therefore, for $z = 8$ and $P = 0.5$ Table 1 summarises the different payoffs for a player depending on the number of C's and D's in its neighbourhood.

Let's start with the peak. For U_{min} greater than 6, only D's surrounded by at least 4 C's can achieve the minimum U_{min} , so cooperation grows dramatically. This corresponds to $\epsilon \lesssim 0.16$, (i.e. $T \lesssim 1.16$). If $U_{min} = 8$ then c drops abruptly since even C agents surrounded entirely by other C's cannot survive anymore.

² The relative error in these averages is about 7 percent.

³ Both for $z=4$ or $z=8$, $P=0.5$ leads to $c=0$ when simulating the model of Ref. [9].

Table 1. Score of a player depending on its state C or D and the number of C and D agents in its $z = 8$ neighbourhood for $T=1+\epsilon$ and $P=0.5$

	8C, 0D	7C, 1D	6C, 2D	5C, 3D	4C, 4D	3C, 5D	2C, 6D	1C, 7D	0C, 8D
C	8	7	6	5	4	3	2	1	0
D	$8.0 + 8\epsilon$	$7.5 + 7\epsilon$	$7.0 + 6\epsilon$	$6.5 + 5\epsilon$	$6.0 + 4\epsilon$	$5.5 + 3\epsilon$	$5 + 2\epsilon$	$4.5 + \epsilon$	4

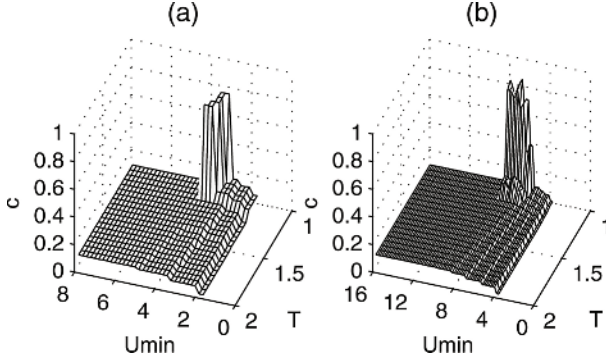


Fig. 1. Fraction of cooperators for $p = 0.1$: (a) $z=4$ neighbours and (b) $z=8$ neighbours

The stepladder structure can be easily explained considering the payoff values for D agents. As long as U_{min} increases each D agent needs more C agents in its surroundings in order to achieve this threshold. So cooperation grows with U_{min} by steps at the values: $U_{min}=T+7P=4.5+\epsilon$, $U_{min}=2T+6P=5+2\epsilon$ and so on, which correspond to straight lines with different slopes in the (T, U_{min}) plane.

Finally, when $U_{min} > 8T$ the minimum required is above any agent's possible score, then the fraction of agents C one time step further will be given by

$$c(t+1) = pf_D + (1-p)f_C, \quad (1)$$

where f_D (f_C) stands for the fraction of agents whose most successful neighbour is a D (C). As none of the agents achieves the threshold, everyone with probability p adopts a state opposite to the one adopted by the most successful neighbour. For relatively small values of p , $f_C \approx 0$ i.e. $f_D \approx 1$ and thus from (1) $c \approx p$. This explains why the height of the plateau coincides with p .

3.2 Hawk-Dove: $P=0$, $S=0.5$, $R=1$, $1 < T \leq 2$ and Stag Hunt: $S=0$, $P=0.5$, $T=1$, $1 < R \leq 2$

Let's briefly review what happens for these other two games. In Fig. 2.a we observe a qualitatively similar 3d plot for c in the case of the H-D. There is a peak in the same place although thicker. On the other hand the SH (see Fig. 2.b) is a much more cooperative than the PD and the threshold has no drastic effects except when R is close to 1. In that case, once again, there are some intermediate values of U_{min} that optimise c .

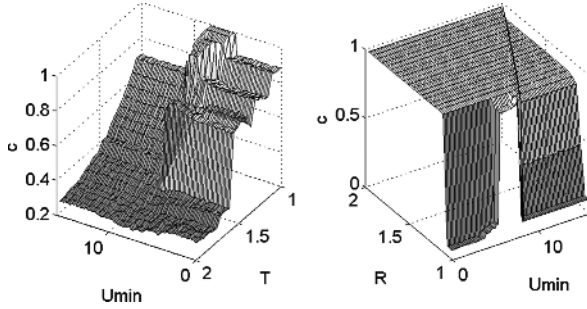


Fig. 2. Fraction of cooperators c for $z=8$ & $p=0.1$. (a) H-D (b) SH.

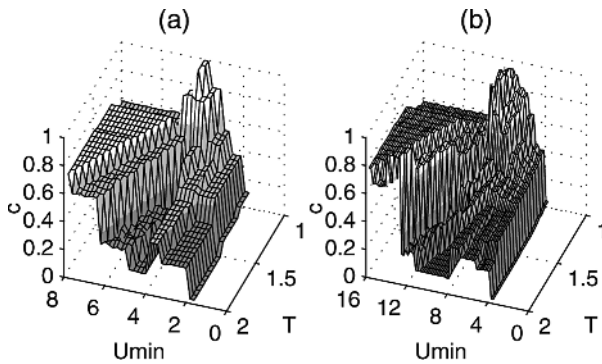


Fig. 3. c for the PD hybrid variant ($p=0.1$). (a) $z=4$ neighbours and (b) $z=8$ neighbours.

3.3 Hybrid Version: Natural Selection Complemented with a Pavlovian Criterion

A relevant input for the behaviour update rule is the comparison of the individual’s score with U_{min} . If it is above U_{min} then the agent’s behaviour may be worth keeping even if it is not the most successful neighbour msn . Then we implement this variant as follows: If the cells’s score is above U_{min} it copies the state of the msn with probability p (or equivalently remains in its state with probability $1 - p$). If the cells’s score is below U_{min} there are two alternatives: If $U^{msn} > U_{min}$ ($U^{msn} < U_{min}$) the cell copies the state of the msn (opposite of the msn) with probability $1 - p$ or remains in its state with probability p .

Therefore, this variant interpolates between the ordinary evolutionary recipe of copying the msn and the ”win-stay, lose-shift” criterion analysed in detail by Herz [11].

The main modifications in the ”landscape” are shown in Fig. 3. Firstly, a strong increase in c , for all the parameter space surrounding the peak zone, is observed. Secondly, most part of the plateau is replaced by steep ”cliffs”.

3.4 Cluster Structure

Spatial patterns were studied for the three games. The most important result deals with the cluster structure of C agents. It turns out that the PD hybrid model variant exhibits power-law scaling for a very reduced region in the plane $T - U_{min}$ in the vicinity of the point $[T = 1.6, U_{min} = 7.5]$ ⁴. The corresponding histogram of size distribution of clusters is shown in Fig. 4 and fits well with a power law with exponent -1.6357 ± 0.0001 . Power laws are the signature of spatial organization of processes into a critical state without any dominating scale.

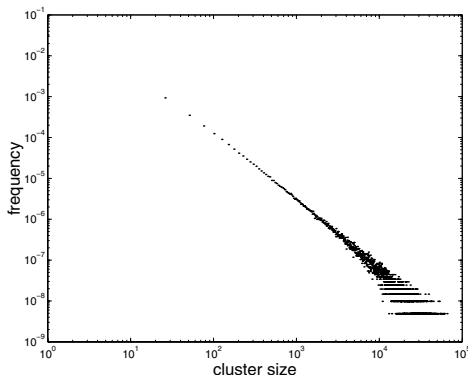


Fig. 4. Dependence of the number of C clusters with size (in logarithmic scale) for the PD hybrid variant for $T = 1.6$, $U_{min} = 7.5$ ($p = 0.1$). (500×500 lattice)

4 Discussion

We extended a simple CA model to include, besides space and time (repetition), the influence of the environment exerting pressure on individuals to cooperate. This is done by requiring a minimum score U_{min} necessary for agents to carry on vital functions.

This recipe gives rise to cooperation among self-interested individuals from evolution in different dilemma games, involving quite arbitrary payoff matrices. Since we resort to the simplest possible agents - unconditional strategists without long term memory - this allows a potential applicability of the model to a wide variety of contexts from virus (recent experiments indicate that RNA virus may engage in simple two-player games [12]) to social sciences problems.

It is remarkable that, for moderate values of the best payoff, there is an intermediate range of values of U_{min} that produces a jump in the cooperation for all the three games analysed.

Finally, we introduced and studied a more complex model in which the ordinary evolutionary recipe, of copying the most successful neighbour, is supplemented with a Pavlovian "win-stay, lose-shift" criterion. Besides leading to global

⁴ The H-D and SH or the basic PD variant don't give rise to power-laws.

optimisation (a much higher cooperation level), this gives rise to power-laws in the size distribution of C-agent clusters. This critical scaling was already found in a different study of CA playing the PD game with Pavlovian strategies [13].

References

1. Axelrod, R. in: *The Evolution of Cooperation*, (Basic Books, New York, 1984);
2. Maynard-Smith, J.: *Evolution and the Theory of Games*, (Cambridge Univ. Press 1982).
3. Flood, M. : Some Experimental Games, Research Memorandum (RAND), RM-789 (1952).
4. Axelrod, R. and Hamilton, W. D. : The evolution of cooperation. *Science* **211** (1981) 1390-1396
5. Axelrod, R. : *J. of Conflict Resolution* **24** (1980) 379.
6. Nowak, M.A. and Sigmund, K. : A strategy of win-stay, lose-shift that outperforms tit for tat in Prisoner's Dilemma, *Nature* 364 (1993) 56-59.
7. Rapoport, A. and Chammah, A.M. :, *Prisoner's Dilemma* (The University of Michigan Press 1965) pp. 73-74.
8. Kraines, D. and Kraines, V. : Pavlov and the Prisoner's Dilemma, *Theory Decision* **26** (1988) 47-79.
9. Nowak, M.A. and May, R. : Evolutionary Games and Spatial Chaos, *Nature* **359** (1992) 826-28.
10. Skyrms, B. : *The Stag Hunt and the Evolution of Social Structure*, Cambridge University Press 2004.
11. Herz, A.V.M. : Collective Phenomena in Spatially Extended Evolutionary Games, *J. Theor. Biol.* **169** (1994) 65-87.
12. Turner, P.E. and Chao, L. : Prisoners Dilemma in an RNA Virus, *Nature* 398 (1999) 441-443.
13. Fort, H. and Viola, S. : Spatial patterns and scale freedom in Prisoner's Dilemma cellular automata with Pavlovian strategies, *J. Stat. Mech.* (2005) P01010.

Coalescing Cellular Automata

Jean-Baptiste Rouquier¹ and Michel Morvan^{1,2}

¹ ENS Lyon, LIP, 46 allée d'Italie, 69364 Lyon, France

² EHESS and Santa Fe Institute

{jean-baptiste.rouquier, michel.morvan}@ens-lyon.fr

Abstract. We say that a Cellular Automata (CA) is coalescing when its execution on two distinct (random) initial configurations in the same asynchronous mode (the same cells are updated in each configuration at each time step) makes both configurations become identical after a reasonable time. We prove coalescence for two elementary rules and show that there exists infinitely many coalescing CA. We then conduct an experimental study on all elementary CA and show that some rules exhibit a phase transition, which belongs to the universality class of directed percolation.

1 Introduction

The *coalescence* phenomenon, as we call it, has been observed for the first time by Nazim Fates [1], in the context of asynchronous cellular automata. Coalescing CA exhibit the following behavior: starting from two different initial random configurations and running the same updating sequence (the same cells are updated at each time step in both configurations), the configurations quickly become identical, i.e. the dynamics not only reach the same attractor, they also synchronize their orbits. This of course appears in trivial situations, for example if the CA converges on a single fixed point, but Nazim Fates has also observed it in a case where the coalescing orbit is absolutely non trivial.

The goal of this paper is to explore this rather strange emergent phenomenon in which the asymptotic behavior seems to be only related to the (random) sequence of update of the cells and not to the initial configuration. This work shows that, in some cases, the randomness used during evolution is as important as the one used during initialization: this stochastic dynamic, with high entropy, is perfectly insensitive to initial condition (no chaos here).

The results presented here are of two kinds. First, we prove the existence of infinitely many different (we precise this notion) non trivial coalescent CA. Secondly, we study by simulation the behavior of all elementary CA (ECA) with regards of this coalescence property, in an asynchronous context in which at each step each cell has a fixed probability α to be updated. We show that over the 88 different ECA, six situations occur: a/ 37 ECA never coalesce; b/ 20 always coalesce in a trivial way (they converge to a unique fixed point); c/ 6 always coalesce on non trivial orbits; d/ 14 combine a/, b/ and c/ depending on α (4 combine a/ and b/, 3 combine b/ and c/, and 7 combine a/ and c/); e/ 7 enter

either full agreement (coalescence) or full disagreement; last, $f/4$ combine $e/$ with either $a/$, $b/$ or $c/$.

We also study the transition between non coalescence and coalescence when α varies for the ECA that combine $a/$ and $c/$: there is a phase transition belonging to the universality class of directed percolation. We thus get a new model of this class, with a few variants. An unusual fact among directed percolation models is that the limit of the sub-critical regime is neither a single absorbing state, nor a set of fixed points, but a non trivially evolving phase. Its originality and links to other domains could help understanding this class and hopefully lead to analytical results.

The paper is organized as follows. Section 2 gives definitions and notations. We prove in section 3 that, under certain conditions, CA 6 and 7 (using the Wolfram’s numbering of ECA) are coalescing and show how to construct from them coalescing CA with arbitrarily many states. We also prove that CA 15 and 170 either coalesce or enter total disagreement, each case occurring with probability $\frac{1}{2}$. In Section 4, we describe the exhaustive simulation study of all ECA and then check the directed percolation hypothesis. Moreover, we prove that some CA exhibit two phase transitions: one for small α and one for high α .

2 Definitions and Notations

In this paper, we consider the dynamics of some CA when they are run on an asynchronous mode. Let us start by defining the synchronisms we work with.

Definition 1. *An asynchronous finite CA is a tuple $(Q, d, V, \delta, n, \mu)$ where*

- Q is the set of states;
- $d \in \mathbb{N}^*$ is the dimension;
- $V = \{v_1, \dots, v_{|V|}\}$, the neighborhood, is a finite set of vectors in \mathbb{Z}^d ;
- $\delta : Q^{|V|} \rightarrow Q$ is the transition rule;
- $n \in \mathbb{N}^*$ is the size;
- $\mathcal{U} := (\mathbb{Z}/n\mathbb{Z})^d$ is the cell space (with periodic boundary condition);
- μ , the synchronism, is a probability measure on $\{0, 1\}^{\mathcal{U}}$.

A configuration specifies the state of each cell, and so is a function $c : \mathcal{U} \rightarrow Q$.

The dynamic is then the following. Let c_t denote the configuration at time t , (c_0 is the initial configuration). Let $\{M_t \mid t \in \mathbb{N}\}$ be a sequence of independent identically distributed random variables with distribution μ . The configuration at time $t + 1$ is obtained by

$$c_{t+1}(z) := \begin{cases} c_t(z) & \text{if } M_t(z) = 0 \\ \delta(c_t(z + v_1), \dots, c_t(z + v_{|V|})) & \text{if } M_t(z) = 1 \end{cases} .$$

In other words, for each cell z , we apply the usual transition rule if $M_t(z) = 1$ and freeze it (keep its state) if $M_t(z) = 0$.

Here are the two synchronisms we use. They are the most natural, even if others are possible (like systematic or alternating sweep, or updating some cells more often, but this require non independent M_t and so a more general formalism). If $x \in \{0, 1\}^{\mathcal{U}}$, let $|x|_1$ be the number of 1 in the coordinates of x .

The Partially Asynchronous Dynamic Let $0 < \alpha \leq 1$. For each cell, we update it with probability α , independently from its neighbors. μ is thus the product measure of Bernoulli distributions: $\mu(x) := \alpha^{|x|_1}(1 - \alpha)^{|x|_0}$ (with $0^0 = 1$). The case $\alpha = 1$ corresponds to the synchronous dynamic.

The Fully Asynchronous Dynamic At each step, we choose one cell and update it. Which defines $\mu(x)$ as $1/n$ if $|x|_1 = 1$ and $\mu(x) := 0$ otherwise.

We now introduce the definition of coalescing CA to formalize the observation of [1]. The principle is to use two initial configurations, and to let them evolve with the *same* outcome of the random variables $\{M_t \mid t \in \mathbb{N}\}$. In other words, we use two copies of the CA, and at each time step, we update the same cells in both copies. This comes down to using the same source of randomness for both copies, like in [2] (on another system) where the authors observe a synchronization.

Definition 2. *An asynchronous finite CA is coalescing if, for any two initial configurations, applying the same sequence of updates leads both configurations to become identical within polynomial expected time (with respect to n).*

Any nilpotent CA (converging toward a configuration where all states are identical) is coalescing if it converges in polynomial time. But there are non nilpotent coalescing CA, which we call non trivial. We now consider only those CA.

In the following, we heavily use the simplest CA, namely the Elementary CA: one dimension ($d = 1$), 2 states ($Q = \{0, 1\}$), nearest neighbors ($V = \{-1, 0, 1\}$). There are $2^8 = 256$ possible rules, 88 after symmetry considerations. We use the notation introduced by S. Wolfram, numbering the rules from 0 to 255.

3 Formal Proof of Coalescence

In this section, we prove that there are infinitely many coalescing CA. For that, we prove the coalescence of two particular CA and show how to build an infinite number of coalescing CA from one of them. An easy way to do that last point would be to extend a coalescing CA by adding states that are always mapped to one states of the original CA, regardless of their neighbors. However, we consider such a transformation to be artificial since it leads to a CA that is in some sense identical. To avoid this, we focus on *state minimal* CA: CA in which any state can be reached. Note that among ECA, only 0 and 255 are not state-minimal.

We first exhibit two state-minimal coalescing CA (proposition 1); then, using this result, deduce the existence of an infinite number of such CA (theorem 1); and finally describe the coalescent behavior of two others ECA (proposition 2).

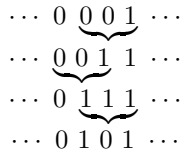
Proposition 1. *Rules 6 and 7 are coalescing for the fully asynchronous dynamic when n is odd.*

Proof. We call *number of zones* the number of patterns 01 in a configuration, which is the number of “blocks” of consecutive 1 (those blocks are the zones). We first consider only one copy (one configuration).

Here is the transition table of 6. Since one cell at a time is updated, and updating the central cell of 101 or 010 does not change its state, zones cannot merge.

Neighbors	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
New state	0	0	0	0	0	1	1	0

On each pattern 111, the central cell can be updated (leading to the pattern 101) before its neighbors (with probability $\frac{1}{3}$) with expected time n . On each pattern 0001, the opposite sequence is possible. It happens without other update of the four cells with probability $1/4^3$ and with an expected time of $3n$. So, as long as there are patterns 000 or 111, the number of zones increases with an expected time $O(n)$. Since there are $O(n)$ zones, the total expected time of this increasing phase is $O(n^2)$.



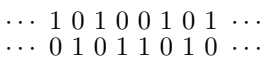
The configuration is then regarded as a concatenation of words on $\{0, 1\}^*$. Separation between words are chosen to be the middle of each pattern 00 and 11, so we get a sequence of words that have no consecutive identical letters, each word being at least two letter long (that is, words of the language “ $(01)^+0? \mid (10)^+1?$ ”). We now show that borders between these words follow a one way random walk (towards right) and meet, in which case a word disappear with positive probability. The CA evolves therefore towards a configuration with only one word.

Updating the central cell of 100 does not change its state, so the borders cannot move towards left more than one cell. On the other hand, updating the central cell of 001 or 110 make the border move. One step of this random walk takes an expected time $O(n)$.

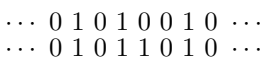
The length of a word also follows a (non-biased) random walk, which reaches 1 after (on average) $O(n^3)$ steps, leading to the pattern 000 or 111. This pattern disappears with a constant non zero probability like in the increasing phase. The expected time for $O(n)$ words to disappear is then $O(n^4)$.

Since n is odd, the two letters at the ends of the words are the same, i.e. there is one single pattern 00 or 11, still following the biased random walk. We now consider again the two copies. This pattern changes the phase in the sequence $(01)^+$, it is therefore a frontier between a region where both configurations agree and a region where they do not. The pattern in the other configuration let us come back to the region where the configurations have coalesced. We study the length of the (single) region of disagreement. It follows a non biased random walk determined by the moves of both patterns.

When this length reaches n , as opposite, the only change happens when the fourth cell is updated, and it decreases the length. So, the random walk cannot indefinitely stay in state n .



On the other hand, when the length reaches 1, one possibility is the opposite, where updating the fifth cell leads to coalescence.



The other possibility is opposite, where updating the $\dots 0 1 0 1 1 0 1 0 \dots$ fifth then the fourth cell leads to the former possibility. $\dots 0 1 0 0 1 0 1 0 \dots$ In each case, coalescence happens with a constant non zero probability. One step of this random walk takes an expected time $O(n)$, the total expected time of the one word step is thus $O(n^3)$ (details on expected time can be found in [3]).

So rule 6 is coalescing. The only difference of 7 is that 000 leads to 010, which does not affect the proof (only the increasing phase is easier). \square

Remark 1. If n is even, the proof is valid until there is only one word, at which point we get a configuration without 00 nor 11. There are two such configurations, if both copies have the same, it is coalescence, otherwise both copies perfectly disagree (definitively). Both happen experimentally.

Theorem 1. *For the fully asynchronous dynamic, there are non trivial state-minimal coalescing cellular automata with an arbitrarily large number of states, and therefore infinitely many non trivial state-minimal coalescing CA.*

Proof. Let \mathcal{A}^2 be the product of a CA $\mathcal{A} = (Q, d, V, \delta, n, \mu)$ by itself, defined as $(Q^2, d, V, \delta^2, n, \mu)$ where $\delta^2((a, b), (c, d), (e, f)) := (\delta(a, c, e), \delta(b, d, f))$. Intuitively, \mathcal{A}^2 is the automaton we get by superposing two configurations of \mathcal{A} and letting both evolve according to δ , but with the same M_t . If \mathcal{A} is state-minimal, so is \mathcal{A}^2 .

Let \mathcal{A} be a coalescing CA. Then \mathcal{A}^2 converges in polynomial expected time towards a configuration of states all in $\{(q, q) \mid q \in Q\}$. From this point, \mathcal{A}^2 simulates \mathcal{A} (by a mere projection of Q^2 to Q) and is therefore coalescing (with an expected time at most twice as long); and so are $(\mathcal{A}^2)^2, ((\mathcal{A}^2)^2)^2, \dots$, etc. We have built an infinite sequence of CA with increasing size. \square

Proposition 2. *15 and 170, for both asynchronous dynamics, coalesce or end in total disagreement, each case with probability 1/2.*

Proof. 170 (shift) means “copy your right neighbor”. The configurations agree on a cell if and only if they agreed on the right neighbor before this cell was updated. So, it is a CA with two states: agree or disagree. This CA is still 170. This rule converges in polynomial time towards 0^* (corresponding to coalescence) or 1^* (full disagreement) [3]. By symmetry, each case has probability 1/2.

15 means “take the state opposed to the one of your right neighbor”, and the proof is identical (the quotient CA is still 170). \square

4 Experimental Study and Phase Transition

In this section, we describe experimental results in the context of partially asynchronous dynamic. We show that many ECA exhibit coalescence and make a finer classification. Specifically, we observe that some ECA undergo a phase transition for this property when α varies. We experimentally show that this phase transition belongs to the universality class of directed percolation.

4.1 Classifying CA with Respect to Coalescence

Protocol. We call *run* the temporal evolution of a CA when all parameters (rule, size, α and an initial configuration) are chosen. We stop the run when the CA has coalesced, or when a predefined maximum running time has been reached.

Let us describe the parameters we used. We set $n = 500$ and $n = 2000$ and got the same results. [3] showed rigorously that α close to 1 (more updates) does not mean faster convergence (indeed, it is proportional to $\frac{1}{\alpha(1-\alpha)}$). We thus repeat each run three times: for $\alpha \in \{0.05, 0.50, 0.95\}$. The maximum number of time steps is equal to a few times n^2 . For each rule, we do 30 runs: 10 random initial configurations (to ensure coherence) times 3 values of α .

Results. We get the following empirical classes of behaviour:

- a/ Some CA never coalesce (or take a too long time to be observed): 4, 5, 12, 13, 25, 28, 29, 33, 36, 37, 41, 44, 45, 51, 54, 60, 72, 73, 76, 77, 78, 90, 94, 104, 105, 108, 122, 132, 140, 142, 150, 156, 164, 172, 200, 204, 232.
- Some CA coalesce rapidly.
- b/ The trivial way to do this is to converge to a unique fixed point. One can consider the two copies independently, and wait for them to reach the fixed point, the CA has then coalesced. This is the case for 0, 2, 8, 10, 24, 32, 34, 38, 40, 42, 56, 74, 128, 130, 136, 138, 152, 160, 162, 168.
- c/ The non trivial rules are 3, 11, 19, 35, 46, 154.
- d/ Some CA combine two of the three previous behaviors, depending on α . 18, 26, 106, 146 combine a/ and b/; 50, 58, 134 combine b/ and c/; 1, 9, 27, 57, 62, 110 and 126 combine a/ and c/ (see fig. 1).
- e/ Some CA end in either full agreement between configurations (coalescence) or full disagreement, depending on the outcome of M_t and the initial configuration: 14, 15, 23, 43, 170, 178, 184.
- f/ 6 combines the previous point (for small α) with b/, 7 do the same but with c/, 22 and 30 combine it (for small α) with a/.

Let us to study the phase transition (coalescence or not) when α changes, that is, rules 1, 9, 27, 57, 62, 110 and 126. We test the hypothesis of directed percolation. Some rules (9, 110, 126) show two phase transitions, one for low α , noted 9_ℓ , one for high α , noted 9_h . The ones for low α (9_ℓ , 110_ℓ and 126_ℓ) are “reversed”, that is, coalescence (sub-critical regime) occurs for higher α . 57 is also reversed.

4.2 Directed Percolation

Due to lack of space, we refer to [4] for a presentation of directed percolation, which also explains *damage spreading*, another point of view on this phenomenon.

Our active sites are the cells where the configurations disagree (density of such sites is written ρ). Percolation appears when varying α , see fig. 1. The aim is thus to identify β assuming that $\rho(\alpha) = c(\alpha - \alpha_c)^\beta$ for some c and α_c .

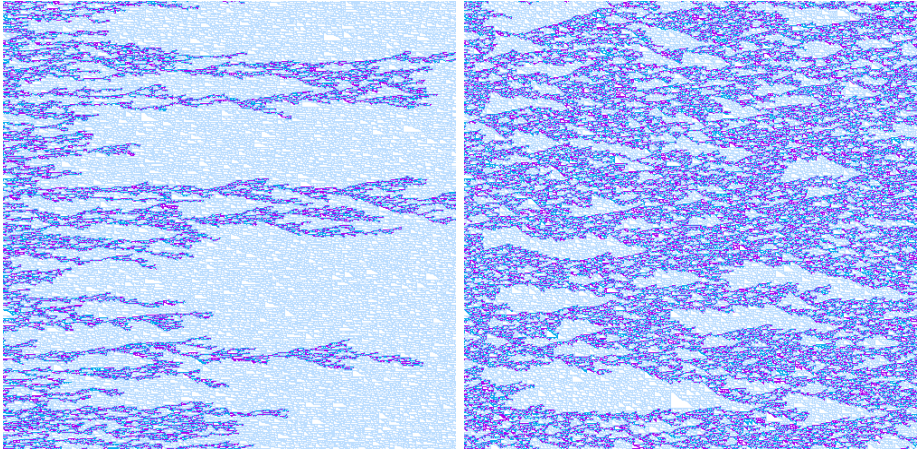


Fig. 1. (color online) rule 110, $n = 500$. Time goes from left to right, during 500 steps. Active sites are dark, coalesced site are light (with light blue standing for state 1, white for 0). Left: sub-critical phase ($\alpha = 0.47 < \alpha_c \simeq 0.566$), branches die. Right: supercritical phase ($\alpha = 0.65 > \alpha_c$), active sites spread.

Measure of α_c . We use the method described in [4]: plot the density ρ of active sites versus time in logarithmic scale and find the α value for which one gets a straight line (for $\alpha < \alpha_c$, the AC coalesce faster, for $\alpha > \alpha_c$, it has a positive asymptotic ρ). We used random initial configuration with each state equiprobable. To get readable plots we needed up to $n = 10^6$ cells and 10^7 time steps. We get (recall that α_c is not universal, it is just used to compute β):

rule	1	9 _l	9 _h	27	62	110 _l	110 _h	126 _l	126 _h	57
$\alpha_c > \dots$	0.102	0.073	0.757	0.856	0.598	0.073	0.566	0.101	0.720	0.749
$\alpha_c < \dots$	0.103	0.074	0.758	0.858	0.599	0.075	0.567	0.102	0.721	0.750

Note that the α_c of 1 and 126_l, like 9_l and 110_l, are very close.

Measure of β . We now plot ρ vs α (fig. 2). The assumption $\rho = c(\alpha - \alpha_c)^\beta$ is valid only near α_c . To determine which points should be taken into account, we plot $\rho(\alpha)$ on a logarithmic scale with x -origin roughly equal to α_c (precision does not affect the result). We keep only the beginning of the curve which is a straight line. We varied the number of points taken into account to estimate the loss of precision due to this choice.

Protocol $n = 10\,000$. We let the system evolve for $T_{tr} = 100\,000$ steps, then measure ρ during $T_{samp} = 10\,000$ step and compute the average. We repeat such a run for each α value with a fine sampling. The fact that the curve is smooth (except very near α_c) tells us that measures do not depend much on the randomness (nor on T_{samp}) and that we do not need error bars. We also checked that the results do not vary when we change n and T_{tr} .

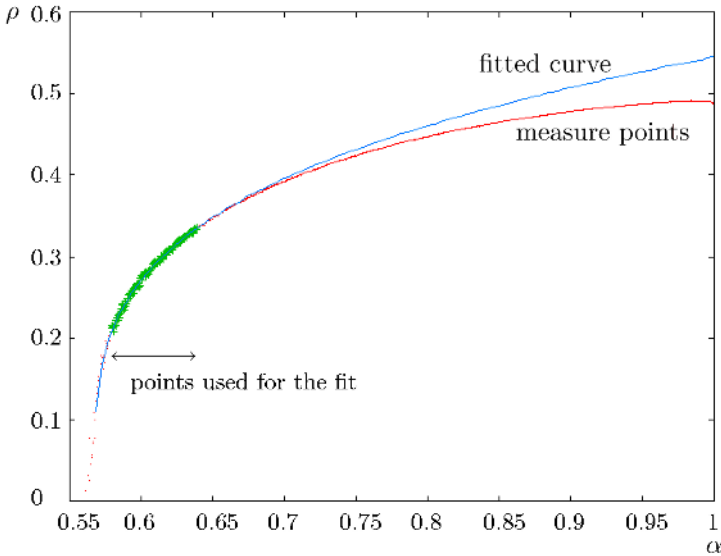


Fig. 2. $\rho(\alpha)$. Rule 110, $n = 10000$. Fitted curve agrees with points used for the fit (bold).

The fit gives the following ranges, taking into account uncertainty about α_c and which points to keep for the fit. Experimental value for β measured on other systems is 0.276.

rule	1	9_ℓ	9_h	27	62	110_ℓ	110_h	126_ℓ	126_h	57
$\beta > \dots$	0.265	0.270	0.273	0.258	0.270	0.270	0.271	0.250	0.260	0.248
$\beta < \dots$	0.279	0.295	0.283	0.305	0.281	0.291	0.281	0.276	0.276	0.281

As expected, this model seems to belong to the universality class of directed percolation (except perhaps 27, due to higher noise and thus lack of precision).

Source code is available on cimula.sf.net.

References

1. Fatès, N., Morvan, M.: An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems* **16** (2005) (to appear).
2. Kaulakys, B., Ivanauskas, F., Mekauskas, T.: Synchronization of chaotic systems driven by identical noise. *International Journal of Bifurcation and Chaos* **9**(3) (1999) 533–539
3. Fates, N., Regnault, D., Schabanel, N., Thierry, E.: Asynchronous behavior of double-quiescent elementary cellular automata. In: *LATIN*. (2006) (to appear).
4. Hinrichsen, H.: Nonequilibrium critical phenomena and phase transitions into absorbing states. *Advances in Physics* (2000) 815

Cellular Automata Architecture for Elliptic Curve Cryptographic Hardware^{*}

Jun-Cheol Jeon, Kee-Won Kim, Byung-Heon Kang, and Kee-Young Yoo^{**}

Department of Computer Engineering, Kyungpook National University,
Daegu, 702-701 Korea
{jcjeon33, nirvana, bhkang}@infosec.knu.ac.kr,
yook@knu.ac.kr

Abstract. Elliptic Curve Cryptosystems (ECC) are in the spotlight due to their significantly smaller parameters. The most costly arithmetic operation in ECC is division, which is performed by multiplying the inverse of a multiplicand. On the other hand, Cellular Automata (CA) have attracted a lot of attention regarding their potential for various applications. Thus, this paper presents an EC-based hardware architectural model for division based on CA over Galois Field $GF(2^n)$. The proposed architectural model is highly regular, expandable, and it has reduced latency based on periodic boundary CA. The proposed architecture can be easily implemented into the hardware design of crypto-coprocessors.

1 Introduction

In cryptography, in order to achieve a high level of security, many public-key algorithms that rely on computations in $GF(2^n)$ require a large fields, and some fields need to be as big as $GF(2^{2000})$. Hence, there is a need to develop an efficient algorithm for the multiplication in $GF(2^n)$. Significantly smaller parameters, however, can be used in ECC than in other competitive systems such RSA and ElGamal, but with an equivalent level of security. The benefits of having smaller key sizes include faster computations and reductions in processing power, storage space, and bandwidth. This makes ECC ideal for constrained environments such as PDAs, cellular phones, and smart cards [1].

ECC was proposed as an alternative to the established public-key cryptosystems of RSA and ElGamal, and it has recently received a great deal of attention in various industries and academia [2, 3]. The main reason for this attention is that there is no sub-exponential algorithm that can solve discrete logarithm problem on a properly chosen elliptical curve. The main operation of ECC is an inverse/division operation, which can be regarded as a special case of exponentiation [4]. However, since a division operation is quite time consuming, efficient algorithms are required for practical applications, especially for a public key cryptosystem where operands can be as large as 512bits or even larger.

^{*} This work was supported by the Brain Korea 21 Project in 2006.

^{**} Corresponding author.

Finite field $GF(2^n)$ arithmetic operations have recently been applied to a variety of fields, including cryptography and error-correcting codes [5]. A number of modern public key cryptography systems and schemes, for example, the Diffie-Hellman key pre-distribution, ElGamal cryptosystem, and ECC, require division and inversion operations [6]. Wang [7] proposed parallel-in parallel-out division architecture with a latency of $(2n^2-1.5n)$ and a critical path of $(T_{2AND}+3T_{2XOR})$. Kim's serial-in serial-out architecture [8] has a latency of $(2n^2-2n)$ and a critical path of $(2T_{2AND}+3T_{2XOR}+T_{MUX})$. However, fast arithmetic architecture is still needed to design dedicated high-speed circuits.

Cellular automata have been used in evolutionary computation for over a decade. They have been used in various applications, such as parallel processing and number theory. CA architecture has been used to design arithmetic computations that Zhang [9] proposed an architecture with programmable cellular automata, Choudhury [10] designed an LSB multiplier based on CA, and Jeon [11] proposed a simple and efficient architecture based on periodic boundary CA.

This paper proposes CA architecture based on EC cryptography for division. We focused on the architecture in ECC, which uses restricted irreducible polynomials, especially, trinomials and pentanomials. The structure has a time complexity of $(n^2-n)(T_{2AND}+T_{2XOR}+T_{MUX})$ and a hardware complexity of $(nAND+(n+2)/(n+6)XOR+nMUX+4nREGISTER)$. In addition, our architecture can easily be expanded to be used in other public key cryptosystem with additional $(n-2)/(n-6)$ XOR gates. Our architecture focuses on both area and time complexity.

The rest of this paper is organized as follows. The theoretical background, including finite fields, ECC, and CA, is described in Section 2. Section 3 presents the proposed division architecture based on CA, and we present our discussion, together with a comparison of the performances between the proposed architecture and previous works, in Section 4. Finally, Section 5 presents our conclusion.

2 Preliminary

In this section, we discuss the mathematical background of the finite field and ECC, and the characteristics and properties of cellular automata.

2.1 Finite Fields

A finite field, which is a set of finite elements, can be defined by commutative law, associative law, distributive law and it contains for facilitates for addition, subtraction, multiplication, and division. A number of architectural models have already been developed to construct low complexity bit-serial and bit-parallel multiplications, by using various irreducible polynomials that can reduce the complexity of modular multiplication. Since a polynomial basis operation does not require a basis conversion, it can be readily matched to any input or output system. In addition, due to its regularity and simplicity, the ability to design and expand it into high-order finite fields, with a polynomial basis, is easier to realize than with other basis operations [12].

A finite field can be viewed as a vector space of dimension n over $GF(2^n)$. That is, there exists a set of n elements $\{1, \alpha, \dots, \alpha^{n-2}, \alpha^{n-1}\}$ in $GF(2^n)$, such that each

$A \in GF(2^n)$ can be written uniquely in the form $A = \sum A_i \alpha^i$, where $A_i \in \{0,1\}$. This section provides one of the most common bases of $GF(2^n)$ over $GF(2)$, which are polynomial bases [12, 13]. Let $f(x) = x^n + \sum_{i=0}^{n-1} f_i x^i$, where $f_i \in \{0,1\}$, for $i = 0, 1, \dots, n-1$, be an irreducible polynomial of degree n over $GF(2)$. For each irreducible polynomial, there exists a polynomial basis representation. In such a representation, each element of $GF(2^n)$ corresponds to a binary polynomial of less than n . That is, for $A \in GF(2^n)$ there exists n numbers $A_i \in \{0,1\}$ such that $A = A_{n-1} \alpha^{n-1} + \dots + A_1 \alpha + A_0$. In many applications, such as cryptography and digital communications, the polynomial basis is still the most widely employed criterion [7, 8, 14]. In the following, we confine our attention to computations that use the polynomial basis.

2.2 Elliptic Curve Cryptosystem

In ECC, computing kP is the most important arithmetic operation, where k is an integer and P is a point on the elliptic curve. This operation can be computed by the addition of two points k times. ECC can be done with at least two types of arithmetic, each of which gives different definitions of multiplication [15]. Two types of arithmetic are, namely, \mathbf{Z}_p arithmetic (modular arithmetic with a large prime p as the modulus) and $GF(2^n)$ arithmetic, which can be done with shifts and exclusive-ors.

We focused on $GF(2^n)$ arithmetic operation. Let $GF(2^n)$ be a finite field by definition. Then, the set of all solutions for equation $E: y^2 + xy = x^3 + a_2x^2 + a_6$, where $a_2, a_6 \in GF(2^n)$, $a_6 \neq 0$, together with the special point called the point at infinity \mathbf{O} , is a non-supersingular curve over $GF(2^n)$. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points in $E(GF(2^n))$ given in affine coordinates [16]. Assume that $P_1, P_2 \neq \mathbf{O}$, and $P_1 \neq -P_2$. The sum $P_3 = (x_3, y_3) = P_1 + P_2$ is computed as follows; if $P_1 \neq P_2$ then $\lambda = (y_1 + y_2)/(x_1 + x_2)$, $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$, $y_3 = (x_1 + x_3)\lambda + x_3 + y_1$, and if $P_1 = P_2$ (called point doubling), then $\lambda = y_1 / (x_1 + x_1)$, $x_3 = \lambda^2 + \lambda + a_2$, $y_3 = (x_1 + x_3)\lambda + x_3 + y_1$ (see [16]).

In either case, the computation process requires one division, one squaring, and one multiplication. Squaring can be substituted by multiplication. From the point addition operation, it should be noted that no computation processes, except for addition, are performed at the same time due to the data dependency. Therefore, the sharing hardware between the division and multiplication processes is more desirable than the separated implementation of the division and multiplication processes [4, 13].

The additive inverse and multiplicative inverses in $GF(2^n)$ can be calculated efficiently using the extended Euclidean algorithm. Division and subtraction are defined in terms of additive and multiplicative inverses: $A-B$ is $A+(-B)$ in $GF(2^n)$ and A/B is $A \cdot (B^{-1})$ in $GF(2^n)$. Here, the characteristic 2 finite fields $GF(2^n)$ used should have $n \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}$ [4]. Addition and multiplication in $GF(2^n)$ should be performed by using one of the irreducible binary polynomials of degree n . This restriction is designed to facilitate interoperability while enabling implementers to deploy efficient implementations that are capable of meeting common security requirements [16].

The rule that is used to pick acceptable reduction polynomials is the followings: If a degree n binary irreducible trinomial, $f(x) = x^n + x^k + 1$, for $n > k \geq 1$ exists, then, the irreducible trinomial with the smallest possible k should be used, otherwise the

degree n binary irreducible pentanomial can be used, $f(x) = x^n + x^{k3} + x^{k2} + x^{k1} + 1$, for $n > k3 > k2 > k1 \geq 1$, with $k3$ being as small as possible, $k2$ being as small as possible for the given $k3$, and $k1$ as small as possible for the given $k3$ and $k2$. These polynomials enable the efficient calculations of field operations.

2.3 Cellular Automata

A CA is a collection of simple cells arranged in a regular fashion. CAs can be characterized by four properties: Cellular geometry, neighborhood specifications, the number of states per cell, and the rules that are used to compute to a new state. The next state of a CA depends on the current state and rules [17].

A one-dimensional cellular automaton consists of a linearly connected array of n cells, each of which takes the value of 0 or 1, and a transition function $f(s)$ on the state configuration, s , with q variables. The value of the cell state s_i is updated in parallel, by using this function in discrete time steps as $s_i^{t+1} = f(s_{i+j}^t)$ where $-r \leq j \leq r$ [10]. The parameter q is usually an odd integer, i.e. $q = 2r+1$, where r is often named the radius of the function $f(s)$; the possible configurations and the total number of rules for the two state CAs, with the radius r of the neighborhood are 2^q and 2^n where $n = 2^q$. A new value of the i th cell is calculated by using the value of the i th cell itself and the values of the r neighboring cells to the right and left of the i th cell.

Furthermore, if the same rule applies to all the cells in a CA, the CA is called a uniform or regular CA, whereas if different rules apply to different cells, it is called a hybrid CA. In addition, in the structure of CAs, the boundary conditions should be taken into consideration since there exists no left neighbor of the leftmost cell and right neighbor of the rightmost cell, among the cells that compose the CA. According to the conditions, they are divided into three types: Null Boundary CA (NBCA), Periodic Boundary CA (PBCA), and Intermediate Boundary CA (IBCA). In this paper, we only consider PBCA, which is mainly used in this area because of their efficient cyclic properties.

We employ the characteristics of PBCA, which is that the left neighbor of the leftmost cell becomes the rightmost cell and they are adjacent to each other. If the next state is determined by 2-bit shift to the left, it can be expressed as $s_i^{t+1} = s_{i-2}^t$ ($0 \leq i \leq n-1$). This means that the next state of the i th cell is changed by the second right neighbor of the current i th cell. The proposed architecture carries out shift operations and modular reductions using the introduced property.

3 CA Architecture for Division

This section presents an A/B architecture based on cellular automata. Finite field division in $GF(2^n)$ can be performed by using multiplication and inverse processes; that is, $A/B = AB^{-1}$, where A and B are the elements of $GF(2^n)$. Here, the multiplicative inverse of the field element B can be obtained by recursive squaring and multiplication, since the field element B can be expressed as

$$B^{-1} = B^{2^n - 2} = (B(B(B \dots B(B(B)^2)^2 \dots)^2)^2)^2 \tag{1}$$

Division also can be easily induced by equation (1).

$$C = AB^{-1} = A(B(B(B \dots B(B(B)^2) \dots)^2)^2)^2$$

The above equation can be generalized as follows:

$$C_0 = B$$

$$C_i = B(C_{i-1})^2 = B^{2^{i+1}-1}, (1 \leq i \leq n-2) \tag{2}$$

$$C_{n-1} = A(C_{n-2})^2 = AB^{2^n-2} = AB^{-1} \tag{3}$$

In equation (2), we compute the C_i value for the $n-2$ clock cycles. Then, we can compute the division results by equation (3). Meanwhile, we assume that $XY = D = (D_{n-1} \dots D_1 D_0)$, where X and Y are the field elements; equation (4) is held for a certain k in a reduction trinomial, i.e. $f(x) = x^n + x^k + 1$.

$$D_{n-2} \cdot x^{n-1} + D_{n-3} \cdot x^{n-2} + \dots + (D_{n-1} \oplus D_{k-1})x^k + \dots + D_1 \cdot x^2 + D_0 \cdot x^1 + D_{n-1} \tag{4}$$

In equation (4), shift operations and modular reduction are performed by a given transition rule and reduction trinomial. The operation shown in equation (4) should be performed twice in order to accomplish C_i for $n-2$ times. Therefore, trinomials and pentanomials based multiplications are computed as the next equations respectively.

$$D_{n-3} \cdot x^{n-1} + D_{n-4} \cdot x^{n-2} + \dots + \gamma x^{k+1} + \delta x^k + \dots + D_0 \cdot x^2 + D_{n-1} \cdot x^1 + D_{n-2}, \tag{5}$$

where $\gamma = D_{n-1} \oplus D_{k-1}$ and $\delta = D_{n-2} \oplus D_{k-2}$

$$D_{n-3} \cdot x^{n-1} + D_{n-4} \cdot x^{n-2} + \dots + \zeta x^{k3+1} + \eta x^{k3} + \dots + \zeta' x^{k2+1} + \eta' x^{k2} + \dots + \zeta'' x^{k1+1} + \eta'' x^{k1} + \dots + D_0 \cdot x^2 + D_{n-1} \cdot x^1 + D_{n-2}, \tag{6}$$

where $\zeta = D_{n-1} \oplus D_{k3-1}$, $\zeta' = D_{n-1} \oplus D_{k2-1}$, $\zeta'' = D_{n-1} \oplus D_{k1-1}$, $\eta = D_{n-2} \oplus D_{k3-2}$, $\eta' = D_{n-2} \oplus D_{k2-2}$, and $\eta'' = D_{n-2} \oplus D_{k1-2}$.

The architectural model based on the above equations (5) and (6) are shown in Fig. 1 and Fig. 2. In terms of the CA concept, both architectural models are based on 5-neighborhood and $s_i^{t+1} = s'_{i-2} (0 \leq i \leq n-1)$, which is also expressed as rule 2863311530.

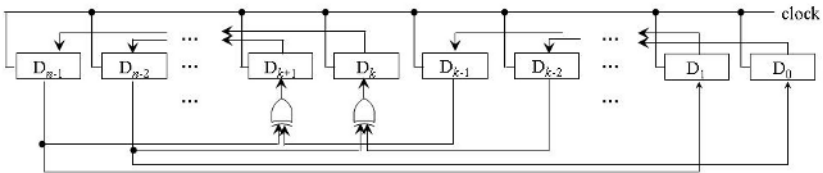


Fig. 1. Multiplication architecture based on irreducible trinomials

In order to satisfy equation (3), A should be multiplied by the square of C_{n-2} . The result is $C = A \cdot B^{-1}$ and when $A = 1$, the algorithm realizes the inverse operation B^{-1} . Fig. 3 shows the division architecture. Each initial value is such that cellular automata have all zeros ($C_i^f = 0, 0 \leq i \leq n-1$); the B register and shift register have B_i values ($B_i^f = B_{n-1} \dots B_2 B_1 B_0$).

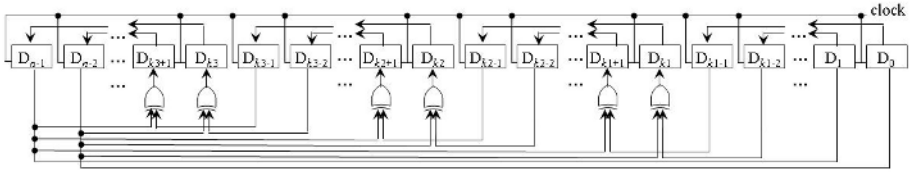


Fig. 2. Multiplication architecture based on irreducible pentanomials

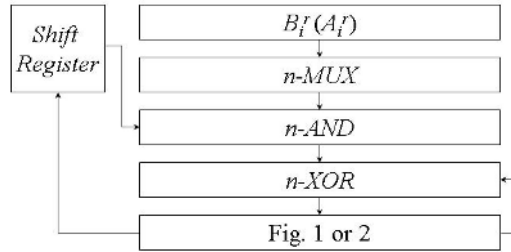


Fig. 3. Division architecture based on Fig. 1 and Fig. 2

For initial n clock cycles, the values in the shift register are sequentially broadcast to the connected line in order to multiply the values in B_i^r , and C_i^r is evolved based on PBCA. After the evolution of C_i^r , the values are transferred to the shift register, just as the initialized form in the shift register. $(C_{n-2})^2$ is computed according to the process described by $n-2$ times.

At the end of the transition, the system chooses A_i values in the B register for the final resultant values. It is possible to perform A/B division in $n(n-1)$ clock cycles by using n AND gates, $n+2n+6$ XOR gates, n Muxes, and $4n$ bits registers. In addition, extra equipment, such as the control signals for transferring results in cellular automata to the shift register and the changing of the values in the B register, right after deriving the values of B^{-1} .

Moreover, our architecture can be easily applied to other public key cryptosystems by using general irreducible polynomials. In Fig. 1 and 2, by using additional $n-2/n-6$ XOR gates, the proposed architecture can perform a general division operation. Although the architecture is used as a general divider, it has the same latency as in Fig. 3 because of the parallelization.

4 Comparison and Analysis

A comparison of the proposed CA architecture for EC-based cryptography, with existing structures, was performed focusing on time and hardware complexity issues. As such, Wang’s [7] and Kim’s [8] division architectural methods were chosen.

Wang proposed a parallel-in parallel-out A/B architectural model, which has a latency of $2n^2-1.5n$ and a critical path of $(T_{2AND}+ 3T_{2XOR})$ over $GF(2^n)$. Kim proposed a serial-in serial-out A/B architectural model, which has a latency of $2n^2-2n$ and critical path of $(2T_{2AND}+ 3T_{2XOR}+ T_{MUX})$ over $GF(2^n)$.

Table 1. A comparison of the performance of the A/B Circuits

Item	Wang et al. [7]	Kim et al. [8]	Fig. 3	
Irreducible polynomial	general	general	Trinomial/ pentanomial	general
Critical path	$T_{2AND} + 3T_{2XOR}$	$2T_{2AND} + 3T_{2XOR} + T_{MUX}$	$T_{2AND} + T_{2XOR} + T_{MUX}$	$T_{2AND} + T_{2XOR} + T_{MUX}$
Latency	$2n^2 - 1.5n$	$2n^2 - 2n$	$2n^2 - n$	$2n^2 - n$
Hardware Complexity	$3n^3 - 3n^2$ AND	$4n^2 - 7n + 3$ AND	n AND	n AND
-Registers(R)	$3n^3 - 3n^2$ XOR	$3n^2 - 5n + 2$ XOR	$n + 2/n + 6$ XOR	$2n$ XOR
-Latch(L)	$8.5n^3 - 8.5n^2$ (L)	$14n^2 - 22n + 8$ (L)	$4n$ (R)	$4n$ (R)
-Inverter(I)		$n^2 - 2n + 1$ Mux	n Mux	n Mux
I/O format	Parallel-in parallel-out	Serial-in serial-out	Serial-in parallel-out	

In general, parallel architectural models need much more hardware equipment than serial fashion architectural models, and the issue of latency is reversed. The proposed architectural model, however, has less complexity than the serial or parallel fashion architectural methods, with respect to both space and time. Our architectural model has been constructed based on ECC. However, our architecture can be easily applied to other public cryptosystems with additional $n-2/n-6$ XOR gates, while the existing systolic architectural models including those of Wang's and Kim's, hardly reduce the level of complexity, although they apply the restrict polynomials.

5 Conclusion

This paper has presented CA architecture in order to compute A/B modulo irreducible trinomials and pentanomials, which are restricted in the Certicom Standard for ECC. We have proposed a simple CA hardware architectural model that is the most costly arithmetic operation scheme in ECC over $GF(2^n)$. The proposed architectural model includes the characteristics of both an evolutionary PBCA and the restricted polynomials, and it has minimized both time and hardware complexity concerns. Moreover, we have shown that our architectural model can be easily applied to general division architectural method with no additional latency needed. Our CA architecture has a regularity and modularity. Accordingly, it can be used as a basic architecture not only for ECC, but also for other public key cryptosystems.

References

1. I. Lopez and R. Dahab, An overview of Elliptic Curve Cryptography, University of Campinas Press, Brazil (2000)
2. N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, Vol. 48. (1987) 203-209
3. V. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology-CRYPTO'85, Springer-Verlog Lecture Notes in Computer Science, Vol. 218. (1986)

4. A. J. Menezes, Elliptic Curve Public Key Cryptosystems, Boston, MA: Kluwer Academic Publishers (1993)
5. T. R. N. Rao and E. Fujiwara, Error-Control Coding for Computer Systems, Englewood Cliffs, NJ: Prentice-Hall (1989)
6. W. Drescher, K. Bachmann, and G. Fettweis, "VLSI Architecture for Non Sequential Inversion over $GF(2^m)$ using the Euclidean Algorithm," The International Conference on Signal Processing Applications and Technology, Vol. 2. (1997) 1815-1819
7. C. L. Wang and J. H. Guo, "New Systolic Arrays for $C+AB^2$, inversion, and division in $GF(2^m)$," IEEE Trans. on Computer, Vol. 49, No. 10. (2000) 1120-1125
8. N. Y. Kim and K. Y. Yoo, "Systolic architecture for inversion/division using AB^2 circuits in $GF(2^m)$," Integration, the VLSI journal, Vol. 35. (2003) 11-24
9. C. N. Zhang, M. Y. Deng, and R. Mason, "A VLSI Programmable Cellular Automata Array for Multiplication in $GF(2^n)$," PDPTA '99 International Conference (1999)
10. P. Pal. Choudhury and R. Barua, "Cellular Automata Based VLSI Architecture for Computing Multiplication and Inverses in $GF(2^m)$," IEEE 7th International Conference on VLSI Design (1994) 279-282
11. Jun-Cheol Jeon and Kee-Young Yoo, "An Evolutionary Approach to the Design of Cellular Automata Architecture for Multiplication in Elliptic Curve Cryptography over Finite Fields," Lecture Notes in Artificial Intelligence PRICAI 2004: Trends in Artificial Intelligence (LNAI 3157), Springer-Verlag, Vol. 3157. (2004) 241-250
12. A. J. Menezes, Applications of Finite Fields, Boston, MA: Kluwer Academic Publishers (1993)
13. IEEE P1363, Standard Specifications for Public Key Cryptography (2000)
14. S. W. Wei, "VLSI architecture of divider for finite field $GF(2^m)$," IEEE International Symposium on Circuit and Systems, Vol. 2. (1998) 482-485
15. C. Kaufman, R. Perlman, and M. Speciner, Network Security private communication in a public world, New Jersey: Prentice Hall (2002)
16. SEC 1: Elliptic Curve Cryptography version 1.0, Certicom Reserch (2000)
17. J. Von Neumann, The theory of self-reproducing automata, University of Illinois Press, Urbana and London (1966)

Efficient Application of Hybrid 150/90 Cellular Automata to Symmetric Cryptography*

A. Fúster-Sabater¹, P. Caballero-Gil², and M.E. Pazo-Robles¹

¹ Instituto de Física Aplicada, C.S.I.C.
Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es

² DEIOC, University of La Laguna
38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. In this work, it is shown that a wide class of nonlinear sequence generators, the so called interleaved sequence generators, can be modelled in terms of linear cellular automata. A simple modelling procedure based on the concatenation of automata has been derived. The cryptographic characteristics of the generated sequences (period, linear complexity, number of different sequences) have been also analyzed. The technique is very simple and can be applied to generators in a range of practical applications.

Keywords: Linear automata, sequence generator, symmetric cryptography.

1 Introduction

Sensitive information to be transmitted or stored requires certain characteristics of confidentiality. Such a quality makes use of an encryption function currently called *cipher* that converts the *plaintext* into the *ciphertext*. Symmetric cryptography are usually divided into two large classes: stream ciphers and block-ciphers. Stream ciphers are the fastest among the encryption procedures so they are implemented in many technological applications e.g. algorithms A5 in GSM communications [8] or the encryption system E0 used in the Bluetooth specifications [1]. Stream ciphers try to imitate the ultimate one-time pad cipher and are designed to stretch a short secret seed (the secret key) into a long sequence (keystream) of seemingly random bits. This keystream is then XORed with the plaintext in order to obtain the ciphertext. Most keystream generators are based on Linear Feedback Shift Registers (LFSRs) [6] combined in a nonlinear way. Such generators produce keystreams with high linear complexity, long period and good statistical properties.

On the other hand, bit sequences generated by one-dimensional binary Cellular Automata (CA) with three site neighborhood and linear transition rules

* Work supported by Ministerio de Educación y Ciencia (Spain) Projects SEG2004-02418 and SEG2004-04352-C04-03.

[2] have been found to be isomorphic to conventional LFSRs. Thus, the latter structures can be simply substituted by the former ones in order to accomplish the same goal: generation of keystreams with application in symmetric cryptography. Nevertheless, the main advantage of these CA is that multiple generators designed in terms of LFSRs as *nonlinear* structures preserve the *linearity* when they are expressed under the form of CA.

The question that arises in a natural way is: are there one-dimensional linear CA able to produce the sequence obtained from any LFSR-based nonlinear generator? The answer is yes and, in fact, this paper stresses the problem of given a particular LFSR-based generator how to find one-dimensional CA that reproduce exactly their output sequences. More precisely, in this work it is shown that a wide class of LFSR-based nonlinear generators, the so called interleaved sequence generators, can be modelled in terms of linear CA. Among this class of generators we can enumerate: Clock-Controlled generators, Cascade-Clock-Controlled generators, Shrinking generator or the generators producing Kasami sequences, GMW sequences, No sequences, Klapper *et al.* sequences ... (see [7]).

Interleaved sequences [7] satisfy a common property: they can be decomposed into a collection of shifts of an unique *PN*-sequence and zero sequences. This kind of sequences can be obtained from linear multiplicative polynomial CA. That is to say, CA made out of a basic structure concatenated a number of times. Therefore, the goal of this work is double:

- To analyze linear multiplicative polynomial CA with emphasis on the cryptographic parameters of their generated sequences (period, linear complexity, characteristic polynomial, number of different output sequences etc.).
- To model the above nonlinear LFSR-based generators in terms of linear multiplicative polynomial CA.

Once the generators have been linearized, all the theoretical background on linear CA can be applied to their analysis and/or cryptanalysis.

2 Fundamentals and Basic Notation

Our attention is concentrated on one-dimensional binary hybrid linear CA with three neighborhood sites and rules 90 and 150. Both rules can be defined as follows:

<p>Rule 90</p> $x_{n+1}^k = x_n^{k-1} + x_n^{k+1}$	<p>Rule 150</p> $x_{n+1}^k = x_n^{k-1} + x_n^k + x_n^{k+1}$
--	---

Indeed, x_{n+1}^k the content of the k -th cell at time $n + 1$ depends on the content of either two different cells (rule 90) or three different cells (rule 150) at time n , for $(k = 1, \dots, L)$ where L is the length of the automaton. A natural form of automaton representation is an L -tuple $\Delta = (d_1, d_2, \dots, d_L)$ where $d_k = 0$ if the k -th cell verifies rule 90 while $d_k = 1$ if the k -th cell k verifies rule 150.

For an one-dimensional null hybrid cellular automaton of length $L = 10$ cells, configuration rules $(90, 150, 150, 150, 90, 90, 150, 150, 150, 90)$ and initial state

Table 1. An one-dimensional null hybrid linear cellular automaton of 10 cells with rule 90 and rule 150 starting at a given initial state

90	150	150	150	90	90	150	150	150	90
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	1	0	1
1	0	0	1	0	0	1	1	0	0
0	1	1	1	1	1	0	0	1	0
1	0	1	1	0	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(0, 0, 0, 1, 1, 1, 0, 1, 1, 0), Table 1 illustrates the formation of its output sequences (binary sequences read vertically) and the succession of states (binary configurations of 10 bits read horizontally). For the previous rules, the different states of the automaton are grouped in closed cycles [5] as well as any output sequence can be generated at any cell provided that the right state cycle is chosen.

Next, some important definitions are introduced.

Definition 1. A *Multiplicative Polynomial Cellular Automaton* is defined as a cellular automaton whose characteristic polynomial is of the form $P_M(X) = (P(X))^p$ where p is a positive integer. If $P(X)$ is a primitive polynomial, then the automaton is called a *Primitive Multiplicative Polynomial Cellular Automaton*.

A widely accepted measure of the unpredictability of a sequence is its *linear complexity* [9] that can be defined as follows:

Definition 2. The *linear complexity* of a sequence is the shorter linear recursion satisfied by such a sequence.

For a binary sequence $\{x_n\}$, the linear recurrence relationship that specifies the n -th element as a combination of the r previous ones can be written:

$$x_n + \sum_{i=1}^r c_i x_{n-i} = 0, \quad n \geq r \tag{1}$$

where the sequence elements (x_n) as well as the coefficients (c_i) belong to a finite field, $GF(2)$. The linear recursion can be expressed as a linear difference equation:

$$(E^r + \sum_{i=1}^r c_i E^{r-i}) x_n = 0, \quad n \geq 0 \tag{2}$$

where E is the shifting operator that operates on x_n , i.e. $E x_n = x_{n+1}$. The characteristic polynomial of the difference equation (2) is:

$$P(X) = X^r + \sum_{i=1}^r c_i X^{r-i}. \tag{3}$$

Let $P(X)$ be a primitive polynomial of degree r and $\alpha \in GF(2^r)$ one of its roots. In this case [9],

$$\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{(r-1)}} \tag{4}$$

are the r roots of such a polynomial. If $P_M(X) = (P(X))^p$, then the roots of $P_M(X)$ will be the same as those of $P(X)$ but with multiplicity p . Consequently, all the binary sequences generated by a primitive multiplicative polynomial cellular automaton will satisfy the linear difference equation:

$$(E^r + \sum_{i=1}^r c_i E^{r-i})^p x_n = 0, \quad n \geq 0. \tag{5}$$

Our analysis focuses on all the possible solutions $\{x_n\}$ of this equation. For a particular sequence generated at the k -th cell, the notation will be $\{x_n^k\}$.

3 Cryptographic Properties of Multiplicative Polynomial CA

In this section, the characteristics of the sequences obtained from this kind of CA are considered in terms of their cryptographic parameters.

3.1 Period of the Generated Sequences

The solutions of the equation (5) are linear combinations of $p \cdot r$ solutions of the form:

$$x_n = \sum_{i=0}^{p-1} \binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \alpha^{2^j n}, \tag{6}$$

where $S_n^i = \sum_{j=0}^{r-1} A_i^{2^j} \alpha^{2^j n}$ represents the n -th element of a PN -sequence [9] of period $2^r - 1$ where $A_i \in GF(2^r)$. Thus, the binary sequence $\{x_n\}$ can be written as the sum of p times the same PN -sequence weighted by a binomial coefficient

$$\{x_n\} = \sum_{i=0}^{p-1} \binom{n}{i} \{S_n^i\}. \tag{7}$$

In addition, each binomial coefficient defines a succession of binary values with a constant period p_i . Therefore, the sequence $\{x_n\}$ is the sum of p sequences of distinct periods $T_i = p_i \cdot (2^r - 1)$ and the period of the total sequence will be:

$$T = \max\{T_i / A_i \neq 0 \ (i = 0, \dots, p - 1)\}. \tag{8}$$

It can be noticed that the period of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in the equation (6). All sequences generated at the same state cycle have the same period. Nevertheless, the same automaton can produce sequences with distinct periods depending on the state cycle considered.

3.2 Linear Complexity of the Generated Sequences

According to [9], the linear complexity of a sequence equals the number and multiplicity of characteristic polynomial roots that appears in the linear recurrence relationship. Therefore, coming back to the equation (6), the linear complexity of $\{x_n\}$ can be computed. In fact, we have r roots each of them with multiplicity p . Thus, if i_{max} is the greatest value of i ($i = 0, \dots, p - 1$) for which $A_i \neq 0$, then the linear complexity LC of the sequence $\{x_n\}$ will be:

$$LC = (i_{max} + 1) \cdot r . \tag{9}$$

It can be noticed that the linear complexity of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in (6). All sequences generated at the same state cycle have the same linear complexity. Nevertheless, the same automaton can produce sequences with distinct linear complexities depending on the state cycle considered.

3.3 The Number of Different Generated Sequences

In order to count the number of different sequences $\{x_n\}$ generated, the choice of the coefficients A_i in the equation (6) must be considered. Different situations can take place:

- If $A_i = 0 \ \forall i$, then all the cells of the cellular automaton will generate the identically null sequence.
- If $A_0 \neq 0$ and $A_i = 0 \ \forall i > 0$, then all the cells of the cellular automaton will generate a unique PN -sequence $\{S_n^0\}$ of period $T_0 = 2^r - 1$ and characteristic polynomial $P(X)$.
- If $A_0 \in GF(2^r)$, $A_1 \neq 0$ and $A_i = 0 \ \forall i > 1$, then there will be $2^r \cdot (2^r - 1)$ possible choices of (A_0, A_1) . According to subsection 3.1, the period of these sequences will be $T_1 = p_1 \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_1 = \frac{2^r \cdot (2^r - 1)}{p_1 \cdot (2^r - 1)} = 2^{r-1} . \tag{10}$$

- In general, if $A_0, A_1, \dots, A_{i-1} \in GF(2^r)$, $A_i \neq 0$ and $A_j = 0 \ \forall j > i$, then there will be $2^{(i \cdot r)} \cdot (2^r - 1)$ possible choices of (A_0, A_1, \dots, A_i) . According to subsection (3.1), the period of these sequences will be $T_i = p_i \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_i = \frac{2^{(i \cdot r)} \cdot (2^r - 1)}{T_i} = \frac{2^{(i \cdot r)}}{p_i} . \tag{11}$$

The total number of different sequences obtained from a multiplicative polynomial cellular automaton will be:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \tag{12}$$

The null sequence is excluded.

4 Constructing Keystream Generators by Concatenation of Primitive CA

In the previous section, structural properties of the sequences obtained from multiplicative polynomial cellular automata have been considered. Now the particular form of these automata is analyzed. Since the characteristic polynomial of such automata is $P_M(X) = (P(X))^p$, it seems quite natural to construct a multiplicative polynomial cellular automaton by concatenating p times the automaton whose characteristic polynomial is $P(X)$. The procedure of concatenation is based on the following result.

Lemma 1. *Let $\Delta = (d_1, d_2, \dots, d_k)$ be the representation of an one-dimensional binary hybrid linear cellular automaton with k cells and characteristic polynomial $P_k(X) = (X + d_1)(X + d_2)\dots(X + d_k)$. The cellular automaton whose characteristic polynomial is $P_{2k}(X) = (P_k(X))^2$ is represented by:*

$$\Delta = (d_1, d_2, \dots, \overline{d_k}, \overline{d_k}, \dots, d_2, d_1). \tag{13}$$

Sketch of proof. The result follows from the fact that [2]:

$$P_{\overline{k}}(X) = P_k(X) + P_{k-1}(X)$$

where $P_{\overline{k}}(X)$ is the polynomial corresponding to $\Delta = (d_1, d_2, \dots, \overline{d_k})$. In the same way

$$\begin{aligned} P_{k+1}(X) &= (X + d_k)P_{\overline{k}}(X) + P_k(X) \\ P_{k+2}(X) &= (X + d_{k-1})P_{k+1}(X) + P_{\overline{k}}(X) \\ &\vdots \\ P_{2k}(X) &= (X + d_1)P_{2k-1}(X) + P_{2k-2}(X). \end{aligned}$$

Thus, by successive substitutions of the previous polynomial into the next one we get:

$$P_{2k}(X) = (X + d_1)P_{2k-1}(X) + P_{2k-2}(X) = (P_k(X))^2. \quad \square$$

In this way, the construction of a linear model based on CA is carried out by the following algorithm:

Input: The parameters of a nonlinear interleaved sequence generator.

- *Step 1:* Determine the irreducible factor $P(X)$ of the characteristic polynomial of the interleaved sequence.
- *Step 2:* Compute the pair of basic CA [2] whose characteristic polynomial is the irreducible factor $P(X)$.
- *Step 3:* For each one of these basic CA, construct by successive concatenations the longer cellular automaton able to generate the original interleaved sequence.

Output: Two linear CA producing the corresponding interleaved sequence.

Let us see a simple example of application. The shrinking generator [4] is a typical example of cryptographic generator with characteristic polynomial $P_M(X)$. The characteristics of this generator can be summarized as follows: The generator is made of two LFSRs, SR1 and SR2, with lengths $L_j(j = 1, 2)$ and feedback polynomials $P_j(X)(j = 1, 2)$, respectively. The decimation rule is: the bit produced by SR2 is discarded if the corresponding bit of SR1 equals 0. The decimated sequence is just the output sequence of the generator. The generator characteristic polynomial is of the form $P_M(X) = (P(X))^p$, $P(X)$ being a primitive polynomial of degree $r = L_2$ and $2^{(L_1-2)} < p \leq 2^{(L_1-1)}$. Moreover, $P(X)$ is the characteristic polynomial of the cyclotomic coset E in $GF(2^{L_2})$ with $E = 2^0 + 2^1 + \dots + 2^{L_1-1}$, see ([6]).

The output sequence will be a solution of a linear difference equation corresponding to primitive multiplicative polynomial CA. Consequently, the shrinking generator can be expressed in terms of a linear model based on CA. A numerical example illustrates the modelling procedure.

Input: A shrinking generator characterized by two LFSRs of lengths $L_1 = 3$, $L_2 = 5$ respectively and characteristic polynomial $P_2(X) = X^5 + X^4 + X^3 + X + 1$. Now, $p = 2^{L_1-1} = 4$ and $r = L_2 = 5$.

Step 1: $P(X)$ the irreducible factor of the generator characteristic polynomial is:

$$P(X) = X^5 + X^3 + 1 .$$

Step 2: The pair of basic CA whose characteristic polynomial is $P(X)$ are:

$$\begin{array}{c} 0\ 1\ 1\ 0\ 0 \\ 0\ 0\ 1\ 1\ 0 \end{array}$$

Step 3: Computation of the required pair of CA by successive concatenations.
For the first automaton:

$$\begin{array}{c} 0\ 1\ 1\ 0\ 0 \\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0 \\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0 \text{ (final automaton)} \end{array}$$

For the second automaton:

$$\begin{array}{c} 0\ 0\ 1\ 1\ 0 \\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \text{ (final automaton)} \end{array}$$

For each automaton, the procedure in *Step 3* has been carried out $L_1 - 1$ times. In fact, each basic complemented automaton has been concatenated $p = 2^{L_1 - 1}$ times.

Output: Two binary strings codifying the required CA.

In this way, we have obtained a pair of linear CA able to produce, among other sequences, the interleaved sequence corresponding to the given shrinking generator. Analogous procedure applies for any of the keystream generators mentioned in section 1.

5 Conclusions

In this work, cryptographic properties of the primitive multiplicative polynomial CA have been analyzed. It is shown that a wide class of LFSR-based sequence generators (interleaved sequence generators) can be described in terms of CA-based structures. In this way, sequence generators conceived and designed as complex nonlinear models can be written in terms of simple linear models. The linearity of these cellular models can be advantageously used in the analysis and/or cryptanalysis of such cryptographic generators.

References

1. Bluetooth, *Specifications of the Bluetooth system*, Version 1.1, February 2001, available at <http://www.bluetooth.com/>
2. K. Cattell *et al.* Synthesis of One-Dimensional Linear Hybrid Cellular Automata, *IEEE Trans. Computers-Aided Design*, Vol. 15, No. 3, pp. 325-335, 1996.
3. S. J. Cho *et al.* Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequences. *Proc. of ACRI 2004. Lecture Notes on Computer Science*, Springer-Verlag, Vol. 3305, pp. 31-39, 2004.
4. D. Coppersmith, H. Krawczyk and Y. Mansour, The Shrinking Generator. *Proc. of CRYPTO'93. Lecture Notes in Computer Science*, Springer Verlag, Vol. 773, pp. 22-39, 1994.
5. A. K. Das, A. Ganguly, A. Dasgupta, S. Bhawmik and P. P. Chaudhuri, Efficient Characterization of Cellular Automata. *IEE Proc.*, Part E. **1**, pp. 81-87, 1990.
6. S.W. Golomb, *Shift Register-Sequences*, Aegean Park Press, Laguna Hill, 1982.
7. G. Gong, Theory and Applications of q-ary Interleaved Sequences, *IEEE Trans. on Information Theory*, Vol. 41, No. 2, pp. 400-411, 1995.
8. GSM, *Global Systems for Mobile Communications*, available at <http://cryptome.org/gsm-a512.htm>
9. E.L. Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, *IEEE Trans. Informat. Theory*, Vol. 22, No. 6, pp. 732-736, 1976.
10. M. Serra *et al.* The Analysis of One-dimensional Linear Cellular Automata and Their Aliasing Properties, *IEEE Trans. on Computer-Aided Design*, Vol. 9, No. 7, pp. 767-778, 1990.
11. X. Sun *et al.* The Concatenation and Partitioning of Linear Finite State Machines, *Int. J. Electronics*. Vol. 78, pp. 809-839, 1995

Cellular Automata Preimages: Count and List Algorithm

Iztok Jeras¹ and Andrej Dobnikar²

¹ iztok.jeras@rattus.info

<http://www.rattus.info/al/al.html>

² University of Ljubljana, Faculty of Computer and Information Science,
Trzaska cesta 25, SI-1001 Ljubljana, Slovenia
andrej.dobnikar@fri.uni-lj.si

Abstract. Preimages of cellular automata are observed. Their number is computed using simple matrix operations. Three algorithms for making a list of preimages are graphically presented using the de Bruijn diagram and its concatenated form: the preimage network.

1 Introduction

Counting of preimages of one-dimensional cellular automata (CA) has been studied by Jen [1] in around 1989, Woorhees [2] in 1993, and McIntosh [7, 8] in 1993. This paper is build on slightly modified methods defined by McIntosh.

The de Bruijn diagram describes the overlapping of strings. It is the main tool for observing CA preimages, but it can graphically represent only preimages of a single cell. In this paper the de Bruijn diagram is firstly redrawn into the preimage diagram, which is then concatenated into the preimage network. This network graphically represents preimages of whole configuration strings.

The listing of preimages is less studied than their counting. In 1992 Wuensche [3, 4, 5] informally described the first algorithm; in this paper it is called the Trace and Backtrack (TB) algorithm. In 2004 Mora, Juárez and McIntosh [6] described a different algorithm that uses the subset diagram (SD) of the de Bruijn diagram. This paper introduces a third method; the Count and List (CL) algorithm. All algorithms are described graphically using the preimage network.

This paper provides only a few specific examples. Documentation and software for a more general and detailed approach can be found in [9].

2 Formal CA Definition

All examples in this paper are based on the rule 110 elementary (2,1) CA (Wolfram notation for closest neighbor boolean CA), so the formal definition focuses on this special case. Only finite CA are observed, since infinite CA may have an infinite number of preimages.

One-dimensional CA are arrays of cells. The value of each cell c_x at position x is either 0 or 1. The state of an array of cells is represented by a *string*

$\alpha = \dots c_{x-1}c_x c_{x+1} \dots$. The string of states of all cells of a finite CA of length N is a *configuration* $C = c_0c_1 \dots c_{N-1}$. The state of the CA at time t is represented by the configuration C^t . In all examples in this paper the so called *ether configuration* $\alpha = 00010011011111_2$ is used (the most common background pattern in rule 110). The subscribed number 2 is used to denote binary strings.

The *neighborhood* $n_x = c_{x-1}c_x c_{x+1}$ of the observed cell is composed of the observed cell and its two closest neighbors (Fig. 1). Neighborhoods of any pair of adjacent cells $c_{x-1}c_x$ overlap for the length of 2 cells. The *overlap* $o_x = c_{x-1}c_x$ uses the position index x from the right cell in the adjacent pair (Fig. 1). Another way to think about overlaps is to observe a cell c_x ; then the *left overlap* is $o_x = c_{x-1}c_x$ and the *right overlap* is $o_{x+1} = c_x c_{x+1}$. Both the left and the right overlap are parts of the neighborhood $n_x = o_x c_{x+1} = c_{x-1} o_{x+1}$. Overlaps are used to represent the boundaries between adjacent segments in cell strings.

At each CA *evolution step* all cells from the current configuration C^t synchronously evolve into the future configuration C^{t+1} . The evolution of every single cell c_x into its future value c_x^{t+1} is defined as the output of the *local transition function* (f), that takes as input the present neighborhood n_x^t of the observed cell at position x . The local transition function is commonly called the *rule*.

$$c_x^{t+1} = f(n_x^t) = f(c_{x-1}^t c_x^t c_{x+1}^t)$$

The evolution step of the whole CA is defined as the *global transition function* $C^{t+1} = F(C^t)$. It is the application of the local transition function to all cells in the configuration simultaneously (Fig. 2).

$$F(c_0c_1 \dots c_{N-1}) = f(n_0)f(n_1) \dots f(n_{N-1})$$

For rule 110 (01101110₂) the local transition function is defined by the next table.

$\underbrace{111}_0$	$\underbrace{110}_1$	$\underbrace{101}_1$	$\underbrace{100}_0$	$\underbrace{011}_1$	$\underbrace{010}_1$	$\underbrace{001}_1$	$\underbrace{000}_0$
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

A *cyclic configuration* is created by joining the left and the right boundary of a finite configuration length N . The position index $x_{\circlearrowleft} = x \bmod N$ becomes a cyclic group of length N . The *cyclic boundary condition* is commonly used since it avoids the explicit definition of the boundary.

A *bounded configuration* is created by cutting a finite configuration length N from an infinite CA. At the left and the right boundary neighborhoods overstep

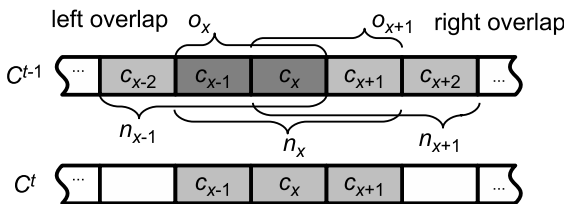


Fig. 1. Neighborhood and overlaps of the observed cell

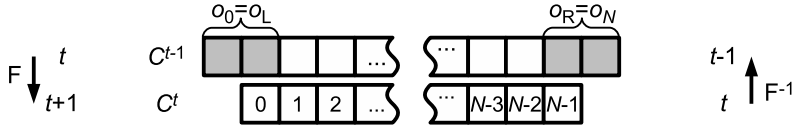


Fig. 2. Configuration boundaries

the configuration by one cell (Fig. 2). To calculate a future configuration C^{t+1} this overstepping cells at both boundaries must be defined. The same problem occurs when calculating preimages, so the preimage C^{t-1} is 2 cells longer than the present configuration C^t . Boundaries are defined by overlaps $o_0 = o_L$ for the left boundary and $o_N = o_R$ for the right boundary.

3 Observing the Past

The *preimages* of a single cell c_x^t are *locally valid neighborhoods* n_x^{t-1} that are mapped into the observed cell value by the local transition function (Fig. 1). The inverse of the local transition function is defined as:

$$f^{-1}(c_x^t) = \{n_x^{t-1} \in \{0, 1\}^3 \mid f(n_x^{t-1}) = c_x^t\}.$$

Preimages C^{t-1} of the present configuration C^t are past configurations that are mapped into the present configuration by the global transition function (Fig. 2). The inverse of the global transition function is:

$$F^{-1}(C^t) = \{C^{t-1} \in \{0, 1\}^N \mid F(C^{t-1}) = C^t\}, \text{ for cyclic configurations, and}$$

$$F^{-1}(C^t) = \{C^{t-1} \in \{0, 1\}^{N+2} \mid F(C^{t-1}) = C^t\}, \text{ for bounded configurations.}$$

Local preimages must overlap correctly to form global preimages. The de Bruijn diagram and the preimage network are graphical representations of overlapping neighborhoods. The first represents preimages of single cells the second preimages of cell strings.

3.1 The de Bruijn Diagram

The *de Bruijn diagram* represents preimages of a single present cell c^t , which are past neighborhoods n^{t-1} (Fig. 3). It is composed of 4 nodes, one for each of the distinct overlaps, and 8 directed links, one for each of the distinct neighborhoods. Nodes are drawn twice and arranged into two identical columns (from overlap 00_2 at the top to overlap 11_2 at the bottom). The two columns can be seen as overlaps (boundaries) at the left and right side of an observed cell (Fig. 1). Directed links connect source nodes o_s^{t-1} (left overlaps) to drain nodes o_d^{t-1} (right overlaps). Links represent neighborhoods $n_{sd}^{t-1} = o_s^{t-1}c_d^{t-1} = c_s^{t-1}o_d^{t-1}$ that can be decomposed into the source or the drain overlap. Links are labeled with the pair n^{t-1}/c^t , where $c^t = f(n^{t-1})$ is defined by the local transition function.

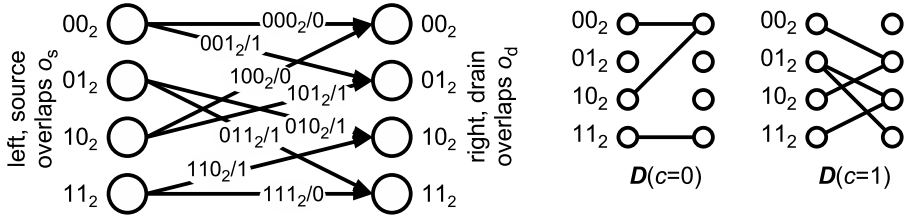


Fig. 3. The de Bruijn diagram (left) and two derived preimage diagrams (right)

The de Bruijn diagram is decomposed into two *preimage diagrams*, one for each of the available cell values. Only *locally valid neighborhoods* (those that are mapped into the observed cell by the local transition function $c^t = f(n^{t-1})$) are allowed to appear in the preimage diagram. All invalid links are removed.

Example: The neighborhood $n^{t-1} = 101_2$ (link) is decomposed into the left overlap 10_2 (source node) and right overlap 01_2 (drain node). Since the neighborhood is translated into $c^t = 0$ its link is labeled $101_2/0$. After the decomposition into preimage diagrams this link becomes part of the diagram for $c = 0$.

The topological matrix of preimage diagrams is used for counting preimages.

Definition 1. The single cell preimage matrix $D(c)$ represents preimages of an observed single present cell with value c . It is a square of 4×4 elements, one for each source-drain overlap pair (o_s, o_d) . The matrix element d_{o_s, o_d} is 1 if first: a past neighborhood $n = o_s c_d = c_s o_d$ exists that can be constructed as a link from the source overlap o_s to the drain overlap o_d ; and second: n is a locally valid neighborhood $f(n) = c$. Else the matrix element is 0.

$$D(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D(1) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The definition of the *single cell preimage matrix* is a special case of the definition of the *cell string preimage matrix*, where $|\alpha| = 1$.

3.2 Preimage Network

Present cells are aligned into a string $\alpha = c_0 c_1 \dots c_{N-1}$ (Fig. 4 bottom). The preimage network of α (Fig. 4 top) is constructed by aligning preimage diagrams for each of the cells in the string side by side in the same order as in the string. Touching node pairs at the same height from two adjacent diagrams are merged into single nodes combining the diagrams into a network. Nodes at the extreme left and extreme right represent the boundaries of the preimage network.

Distinct preimages have distinct representations as paths in the network. Each preimage C^{t-1} of a present configuration C^t is a *globally valid path* that must begin at the left boundary, pass a single link for each present cell, and end at

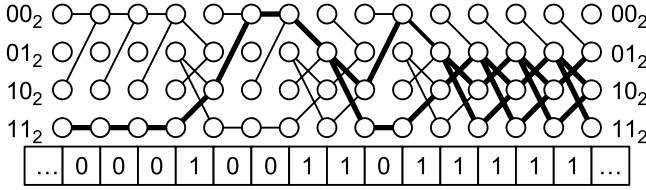


Fig. 4. Preimage network for the bounded ether conf. (globally valid paths are bold)

the right boundary. For cyclic configurations the boundaries are connected, so paths must begin and end at the same overlap $o_L = o_R$. A preimage matrix can be defined to describe the preimage network.

Definition 2. $D(\alpha)$ is the cell string preimage matrix of the observed present string α of length $|\alpha| = N \geq 0$. Elements d_{o_L, o_R} in the preimage matrix represent the number of preimages (distinct paths in the network) that begin at an overlap o_L at the left and end at an overlap o_R at the right boundary (Fig. 2 and 4).

The matrix of an empty string $\alpha = \varepsilon$, $|\alpha| = 0$, is an identity matrix $D(\varepsilon) = I$.

Theorem 1. The cell string preimage matrix $D(\alpha)$ of the string $\alpha = c_0c_1 \dots c_{N-1}$ is the product of the chain of single cell preimage matrices $D(c_x)$.

$$D(\alpha) = \prod_{x=0}^{N-1} D(c_x) = D(c_0)D(c_1) \dots D(c_{N-1})$$

The theorem’s proof is a simple induction on the length of the string. The following preimage matrix describes the ether configuration string.

$$D(\alpha = 00010011011111_2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 \end{bmatrix}.$$

4 Counting Preimages

The number of preimages for bounded configurations $p_{L \leftrightarrow R}$ is computed by applying boundary vectors \mathbf{b}_L and \mathbf{b}_R (unrestricted boundaries in the example) to the preimage matrix of the observed string $D(\alpha)$.

$$p_{L \leftrightarrow R}(\alpha = 00010011011111_2) = \mathbf{b}_L D(\alpha) \mathbf{b}_R^T = [1 \ 1 \ 1 \ 1] D(\alpha) [1 \ 1 \ 1 \ 1]^T = 7$$

The number of preimages for cyclic configurations p_{\circlearrowleft} is the sum of diagonal elements in $D(\alpha)$. This are preimages that begin and end with the same overlap.

$$p_{\circlearrowleft}(\alpha = 00010011011111_2) = 2$$

5 Listing Preimages

5.1 Trace and Backtrack Algorithm

The TB algorithm traces the preimage network in the same way as the wall follower algorithm [10] for solving mazes (Fig. 5). The algorithm starts tracing at left boundary nodes (start-points). At each fork the leftmost (uppermost) path is taken first. If the algorithm reaches a dead-end, it backtracks the traced path to the last fork. Each time the right boundary is reached, the traced path is written into the preimage stack. When all the start-points are exhausted, the algorithm ends, and the stack contains the list of all globally valid preimages.

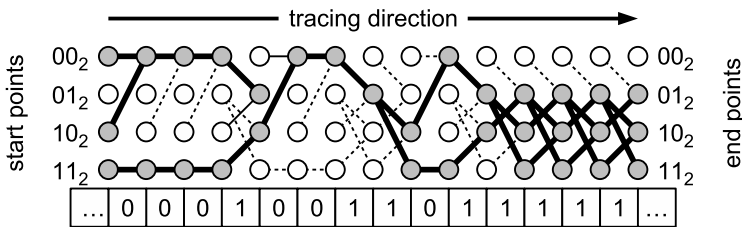


Fig. 5. Tracing and backtracking path on the network of the bounded ether conf.

5.2 Subset Diagram Algorithm

The idea is to trace only paths that do not lead to dead-ends, this are exactly the paths that can be reached from the right boundary (Fig. 6). The network is analyzed (searched for traceable paths) from the right and then traced from the left. Network analysis methods are the same as in the CL algorithm (described next), with the difference, that boolean multiplication (AND operator) is used.

Globally valid paths connect both boundaries (Fig. 4), they are composed of links that can be reached from both the left and the right boundary (links bold on both Fig. 5 and Fig. 6).

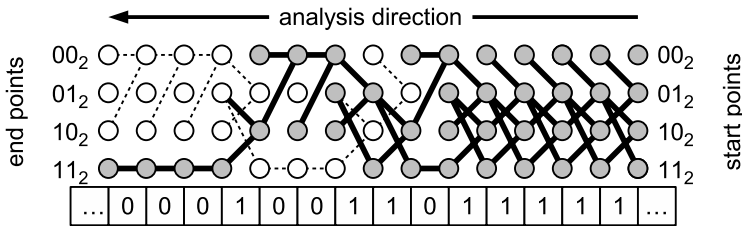


Fig. 6. Paths reachable from the right boundary

5.3 Count and List Algorithm

A further modification to the algorithm is to count paths that can be traced from the right boundary instead of only checking if there are any. Path counters (Table 1) are computed iteratively for all position indexes, starting from counters at the right boundary. For bounded configurations counters are vectors \mathbf{b}_x , starting with the right boundary vector \mathbf{b}_R (unrestricted boundary in the example). For cyclic configurations counters must be matrices \mathbf{D}_x , starting with an identity matrix.

Bounded configurations:

$$\mathbf{b}_N = \mathbf{b}_R = [1, 1, 1, 1]$$

$$\mathbf{b}_x = D(c_x \dots c_{N-1}) \mathbf{b}_R = D(c_x) \mathbf{b}_{x+1}$$

$$\mathbf{b}_0 = D(\alpha) \mathbf{b}_R$$

Cyclic configurations:

$$D_N = I$$

$$D_x = D(c_x \dots c_{N-1}) = D(c_x) D_{x+1}$$

$$D_0 = D(\alpha)$$

Table 1. Counter vectors for the bounded ether configuration

o	counter vectors \mathbf{b}_x														
00	0	0	0	0	7	7	7	0	4	4	3	2	2	1	1
01	0	0	0	7	0	0	4	7	0	5	4	3	2	2	1
10	0	0	0	0	7	7	7	0	4	4	3	2	2	1	1
11	7	7	7	7	0	0	0	4	3	3	2	2	1	1	1
α	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1
x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

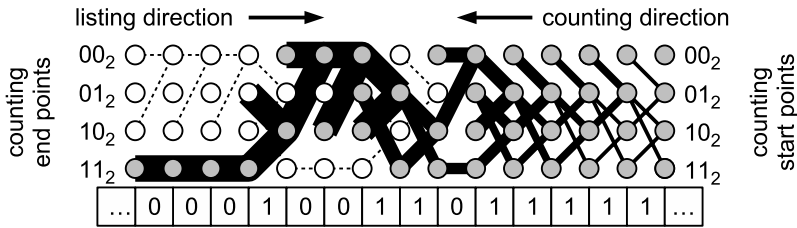


Fig. 7. CL algorithm on the preimage network of the bounded ether conf.

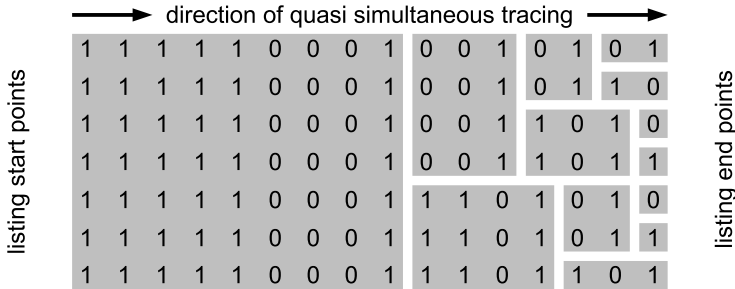


Fig. 8. The list of preimages of the bounded ether conf. as produced by the CL alg.

The computed counters can be presented on the preimage network as link widths (Fig. 7). Logarithmic widths can be used if counters become large.

In the CL algorithm all preimages can be listed quasi simultaneously. The preimage path starts at the left boundary as a wide root, it forks into thinner branches (Fig. 8) and ends as single preimage leaves at the right boundary.

6 Conclusion

The purpose of the research described in the paper was to find a quantitative and qualitative measurement for information losses in irreversible CA. The research is not finished yet, but the logarithm of the number of preimages could be used as a quantitative measurement, and the preimage network could be used to observe how are losses spread in space. The source code for the TB and CL algorithms can be found at <http://www.rattus.info/al/al.html>.

References

1. Jen, E.: Enumeration of Preimages in Cellular Automata. *Complex Systems* **3** (1989) 421-456
2. Voorhees, B.: Predecessors of cellular automata states II. Pre-images of finite sequences. *Physica D* **73** (1993) 136-151
3. Wuensche, A., Lesser, M.: *The Global Dynamics of Cellular Automata*. Addison-Wesley (1992) <http://www.cogs.susx.ac.uk/users/andywu/gdca.html>
4. Wuensche, A.: *Attractor Basins of Discrete Networks*. Cognitive Science Research Paper 461. Univ. of Sussex. D.Phil thesis (1997) ftp://ftp.cogs.susx.ac.uk/pub/users/andywu/papers/aw_thesis.pdf
5. Wuensche, A.: Classifying Cellular Automata Automatically: Finding gliders, filtering, and relating space-time patterns, ... *COMPLEXITY* **4** (1999) 47-66 <ftp://ftp.cogs.susx.ac.uk/pub/users/andywu/papers/cplex.pdf>
6. Mora, J. C. S. T., Juárez, G., McIntosh, H. V.: Calculating ancestors in one-dimensional cellular automata. *International Journal of Modern Physics C* **15** (2004) 1151-1169
7. McIntosh, H. V.: *Linear Cellular Automata Via de Bruijn Diagrams* (1994) <http://delta.cs.cinvestav.mx/~mcintosh/newweb/marcodebruijn.html>
8. McIntosh, H. V.: *Ancestors: Commentaries on The Global Dynamics of Cellular Automata by Andrew Wuensche and Mike Lesser* (1993) <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/wand1/wand1.html>
9. Jeras, I., Dobnikar, A.: Algorithms for Computing Preimages of Cellular Automata Configurations. Submitted for publication to *Physica D* (2005) <http://www.rattus.info/al/al.html>
10. Pullen W. D.: *Maze Algorithms*. www.astrolog.org/labyrnth/algrithm.htm

Self-synchronization of Cellular Automata: An Attempt to Control Patterns

J.R. Sánchez¹ and R. López-Ruiz²

¹ Fac. Ingeniería, Universidad Nacional de Mar del Plata,
Justo 4302, 7600 Mar del Plata, Argentine
jsanchez@fi.mdp.edu.ar

² Department of Computer Science and BIFI, Facultad de Ciencias,
Universidad de Zaragoza, 50009 - Zaragoza, Spain
rilopez@unizar.es

Abstract. Cellular automata display configurations that are constant in time. We implement a stochastic synchronization between the present configurations of the system and its precedent ones in order to search for these constant patterns. For most of the known evolution rules with complex behavior a dynamic competition among all the possible constant patterns is established and no stationary regime is reached. For the particular rule coded by the decimal number 18, a self-synchronization phenomenon can be obtained, even when strong modifications to the synchronization method are applied.

1 Introduction

Cellular automata (CA) are discrete dynamical systems, discrete both in space and time. The simplest one dimensional version of a cellular automaton is formed by a lattice of N sites or cells, numbered by an index $i = 1, \dots, N$, and with periodic boundary conditions. In each site, a local variable σ_i taking a binary value, either 0 or 1, is assigned. The binary string $\sigma(t)$ formed by all sites values at time t represents a configuration of the system. The system evolves in time by the application of a rule Φ . A new configuration $\sigma(t + 1)$ is obtained under the action of the rule Φ on the state $\sigma(t)$. Then, the evolution of the automata can be written as

$$\sigma(t + 1) = \Phi [\sigma(t)]. \quad (1)$$

If coupling among nearest neighbors is used, the value of the site i , $\sigma_i(t + 1)$, at time $t + 1$ is a function of the value of the site itself at time t , $\sigma_i(t)$, and the values of its neighbors $\sigma_{i-1}(t)$ and $\sigma_{i+1}(t)$ at the same time. Then, the local evolution is expressed as

$$\sigma_i(t + 1) = \phi(\sigma_{i-1}(t), \sigma_i(t), \sigma_{i+1}(t)), \quad (2)$$

being ϕ a particular realization of the rule Φ . For such particular implementation, there will be 2^3 different local input configurations for each site and, for each one of them, a binary value can be assigned as output. Therefore there will be 2^8

different rules ϕ , also called the *Wolfram rules*. Each one of these rules produces a different dynamical evolution. In fact, dynamical behavior generated by all 256 rules were already classified in four generic classes. The reader interested in the details of such classification is addressed to the original reference [1].

CA provide us with simple dynamical systems, in which, we would like to find different methods of synchronization. A stochastic synchronization technique was introduced by Morelli and Zanette [2] that works in synchronizing two CA evolving under the same rule Φ . The two CA are started from different initial conditions and they are supposed to have partial knowledge about each other. In particular, the CA configurations, $\sigma^1(t)$ and $\sigma^2(t)$, are compared at each time step. Then, a fraction p of the total different sites are made equal (synchronized). The synchronization is stochastic since the location of the sites that are going to be equal is decided at random. Hence, the dynamics of the two coupled CA, $\sigma(t) = (\sigma^1(t), \sigma^2(t))$, is driven by the successive application of two operators:

1. the deterministic operator given by the CA evolution rule Φ , $\Phi[\sigma(t)] = (\Phi[\sigma^1(t)], \Phi[\sigma^2(t)])$, and
2. the stochastic operator Γ_p that produces the result $\Gamma_p[\sigma(t)]$, in such way that, if the sites are different ($\sigma_i^1 \neq \sigma_i^2$), then Γ_p sets both sites equal to σ_i^1 with the probability $p/2$ or equal to σ_i^2 with the same probability $p/2$. In any other case Γ_p leaves the sites unchanged.

Therefore the temporal evolution of the system can be written as

$$\sigma(t + 1) = (\Gamma_p \circ \Phi)[\sigma(t)] = \Gamma_p[(\Phi[\sigma^1(t)], \Phi[\sigma^2(t)])]. \tag{3}$$

A simple way to visualize the transition to synchrony can be done by displaying the evolution of the difference automaton (DA),

$$\delta_i(t) = | \sigma_i^1(t) - \sigma_i^2(t) | . \tag{4}$$

The mean density of active sites for the DA

$$\rho(t) = \frac{1}{N} \sum_{i=1}^N \delta_i(t), \tag{5}$$

represents the Hamming distance between the automata and verifies $0 \leq \rho \leq 1$. The automata will be synchronized when $\lim_{t \rightarrow \infty} \rho(t) = 0$. As it has been described in [2] that two different dynamical regimes, controlled by the parameter p , can be found in the system behavior:

$$\begin{aligned} p < p_c &\rightarrow \lim_{t \rightarrow \infty} \rho(t) \neq 0 \text{ (no synchronization),} \\ p > p_c &\rightarrow \lim_{t \rightarrow \infty} \rho(t) = 0 \text{ (synchronization),} \end{aligned}$$

being p_c the parameter for which the transition to the synchrony occurs. When $p \lesssim p_c$ complex structures can be observed in the DA time evolution. In Fig. 1, typical cases of such behavior are shown near the synchronization transition. Lateral panels represent both CA evolving in time where the central strip displays

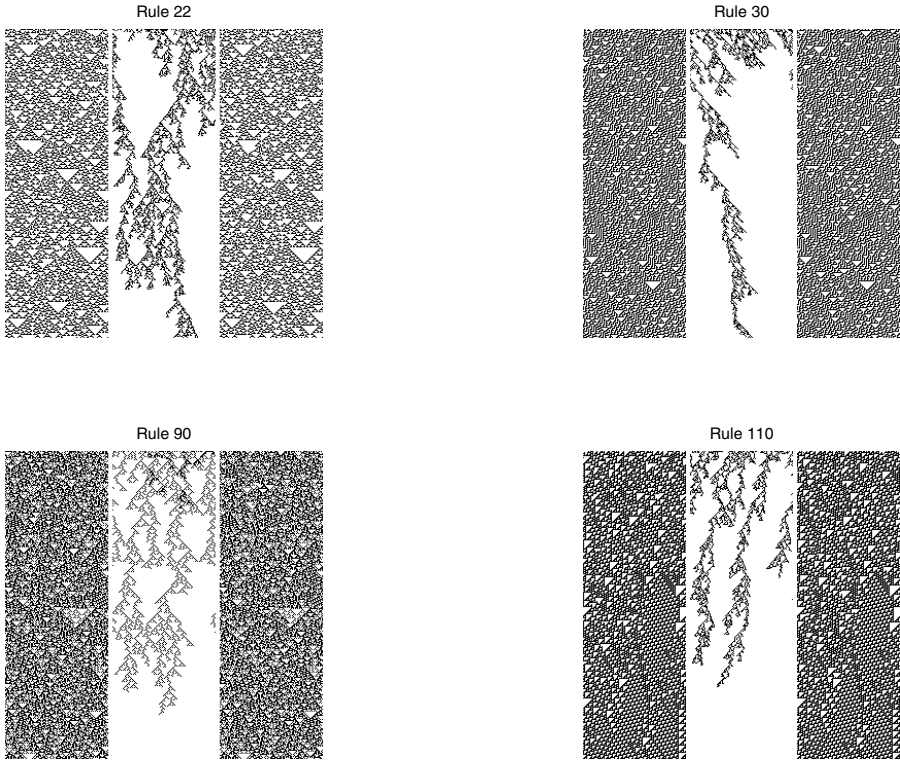


Fig. 1. Spatio-temporal patterns near the synchronization transition. Lateral strips represent both CA evolving in time where the central strip displays the evolution of the corresponding DA. Time goes from top to bottom. Lattice size is $N = 100$ and the number of iterations is $T = 250$. The stochastic coupling between both CA is $p = 0.23$.

the evolution of the corresponding DA. When p comes close to the critical value p_c the evolution of $\delta(t)$ becomes rare and resembles the problem of structures trying to percolate in the plane [3]. A method to detect this kind of transition, based in the calculation of a statistical measure of complexity for patterns, has been proposed in the Refs. [4].

2 Self-synchronization of Cellular Automata

2.1 First Self-synchronization Method

Let us now take a single cellular automaton [5, 6]. If $\sigma^1(t)$ is the state of the automaton at time t , $\sigma^1(t) = \sigma(t)$, and $\sigma^2(t)$ is the state obtained from the application of the rule Φ on that state, $\sigma^2(t) = \Phi[\sigma^1(t)]$, then the operator Γ_p can be applied on the pair $(\sigma^1(t), \sigma^2(t))$, giving rise to the evolution law

$$\sigma(t + 1) = \Gamma_p[(\sigma^1(t), \sigma^2(t))] = \Gamma_p[(\sigma(t), \Phi[\sigma(t)])]. \tag{6}$$

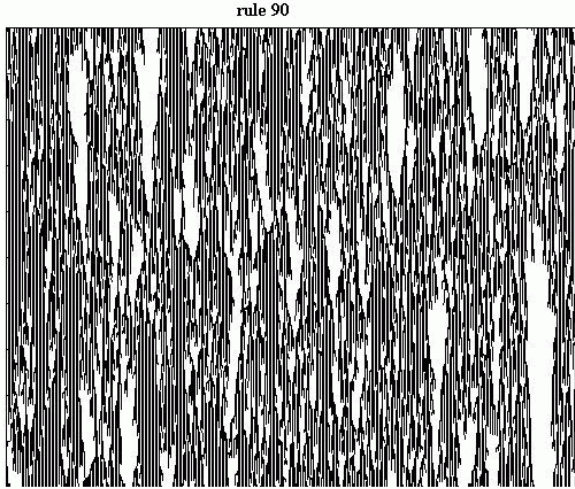


Fig. 2. Rule 90 has two stable patterns: one repeats the 011 string and the other one the 00 string. Such patterns are reached by the first self-synchronization method but there is a dynamical competition between them. In this case $p = 0.9$. Binary value 0 is represented in white and 1 in black. Time goes from top to bottom.

The application of the Γ_p operator is as follows. When $\sigma_i^1 \neq \sigma_i^2$, the sites i of the state $\sigma^2(t)$ are updated to the correspondent values taken in $\sigma^1(t)$ with a probability p . The updated array $\sigma^2(t)$ is the new state $\sigma(t + 1)$.

It is worth to observe that if the system is initialized with a configuration constant in time for the rule Φ , $\Phi[\sigma] = \sigma$, then this state σ is not modified when the dynamic equation (6) is applied. Hence the evolution will produce a pattern constant in time. However, in general, this stability is marginal. A small modification of the initial condition gives rise to patterns variable in time. In fact, as the parameter p increases, a competition among the different marginally stable structures takes place. The dynamics drives the system to stay close to those states, although oscillating continuously and randomly among them. Hence, a complex spatio-temporal behavior is obtained. Some of these patterns can be seen in Fig. 2. However, in rule 18, the pattern becomes stable and, independently of the initial conditions, the system evolves toward this state, which is the null pattern in this case.

2.2 Second Self-synchronization Method

Now we introduce a new stochastic element in the application of the operator Γ_p . To differentiate from the previous case we call it $\tilde{\Gamma}_{\tilde{p}}$. The action of this operator consists in applying at each time the operator Γ_p , with p chosen at random in the interval $(0, \tilde{p})$. The evolution law of the automaton is in this case:

$$\sigma(t + 1) = \tilde{\Gamma}_{\tilde{p}}[(\sigma^1(t), \sigma^2(t))] = \tilde{\Gamma}_{\tilde{p}}[(\sigma(t), \Phi[\sigma(t)])]. \tag{7}$$

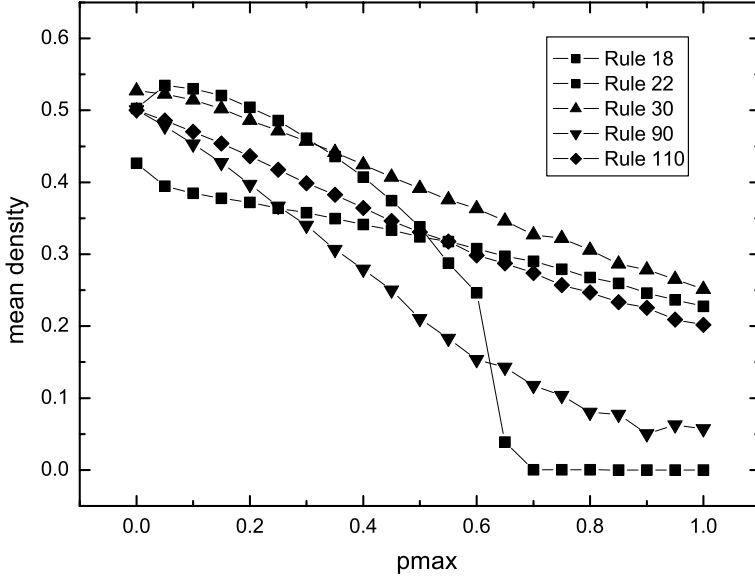


Fig. 3. Mean density ρ vs. $pmax = \tilde{p}$ for different rules evolving under the second synchronization method. The existence of a transition to a synchronized state can be clearly observed for rule 18.

The DA density between the present state and the previous one, defined as $\delta(t) = |\sigma(t) - \sigma(t-1)|$, is plotted as a function of \tilde{p} for different rules Φ in Fig. 3. Only when the system becomes self-synchronized there will be a fall to zero in the DA density. Let us observe again that the behavior reported in the first self-synchronization method is newly obtained in this case. Rule 18 undergoes a phase transition for a critical value of \tilde{p} . For \tilde{p} greater than the critical value, the method is able to find the stable structure of the system. For the rest of the rules the freezing phase is not found. The dynamics generates patterns where the different marginally stable structures randomly compete. Hence the DA density decays linearly with \tilde{p} (see Fig. 3).

2.3 Third Self-synchronization Method

At last, we introduce another type of stochastic element in the application of the rule Φ . Given an integer number L , the surrounding of site i at each time step is redefined. A site i_l is randomly chosen among the L neighbors of site i to the left, $(i-L, \dots, i-1)$. Analogously, a site i_r is randomly chosen among the L neighbors of site i to the right, $(i+1, \dots, i+L)$. The rule Φ is now applied on the site i using the triplet (i_l, i, i_r) instead of the usual nearest neighbors of the site. This new version of the rule is called Φ_L , being $\Phi_{L=1} = \Phi$. Later the operator Γ_p acts in identical way as in the first method. Therefore, the dynamical evolution law is:

$$\sigma(t+1) = \Gamma_p[(\sigma^1(t), \sigma^2(t))] = \Gamma_p[(\sigma(t), \Phi_L[\sigma(t)])]. \quad (8)$$

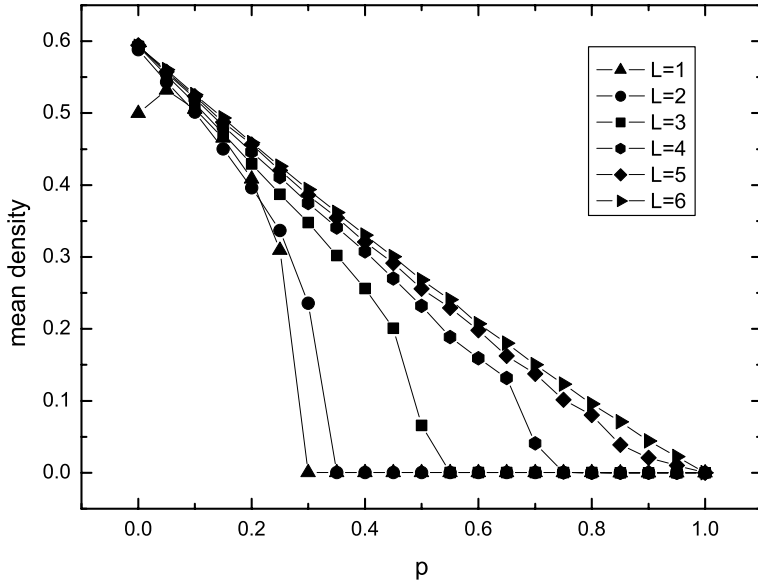


Fig. 4. Mean density ρ vs. p for rule 18 evolving under the third self-synchronization method. The existence of a transition to a synchronized state can be observed despite of the randomness in the election of neighbors within a range L , up to $L = 4$.

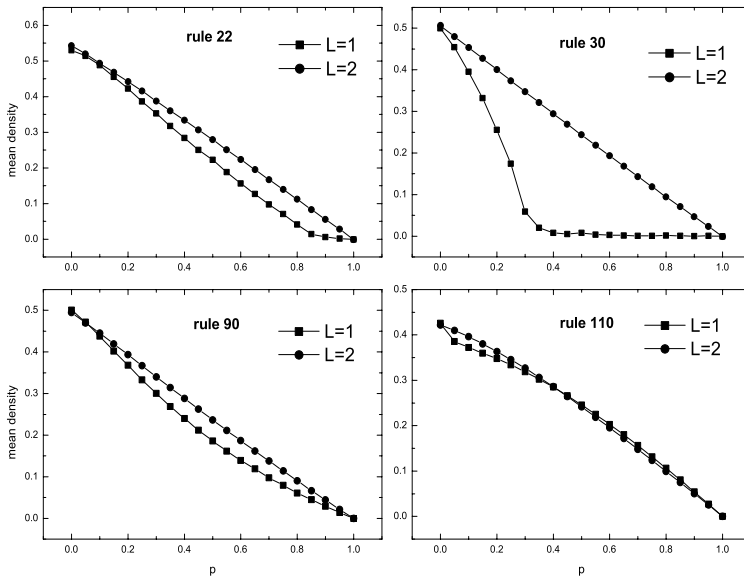


Fig. 5. Mean density ρ vs. p for different rules evolving under the third self-synchronization method. The density of the system decreases linearly with p .

The DA density as a function of p is plotted in Fig. 4 for the rule 18 and in Fig. 5 for other rules. It can be observed again that the rule 18 is a singular case that, even for different L , maintains the memory and continues to self-synchronize. It means that the influence of the rule is even more important than the randomness in the election of the surrounding sites. The system self-synchronizes and decays to the corresponding stable structure. Contrary, for the rest of the rules, the DA density decreases linearly with p even for $L = 1$ as shown in Fig. 5. The systems oscillate randomly among their different marginally stable structures as in the previous methods.

3 Conclusions

Inspired in stochastic synchronization methods for CA, different schemes for self-synchronization of a single automaton have been proposed and analyzed in this work. Self-synchronization of a single automaton can be interpreted as a strategy for searching and controlling the structures of the system that are constant in time. In general, it has been found that a competition among all such structures is established, and the system ends up oscillating randomly among them. However, rule 18 is a unique position among all rules because, even with random election of the neighbors sites, the automaton is able to reach the configuration constant in time.

References

1. Wolfram, S.: Statistical mechanics of cellular automata. *Rev. Mod. Phys.* **55** (1983) 601-644
2. Morelli, L.G., Zanette, D.H.: Synchronization of stochastically coupled cellular automata. *Phys. Rev. E* **58** (1998) R8-R11 ; Synchronization of coupled extended dynamical systems: a short review. *Int. J. Bifurcation and Chaos* **13** (2003) 1-16
3. Pomeau, Y.: Front motion, metastability and subcritical bifurcations in front motion. *Physica D* **23** (1986) 3-11
4. Sánchez, J.R., López-Ruiz, R.: A method to discern complexity in two-dimensional patterns generated by coupled map lattices. *Physica A* **355** (2005) 633-640 ; Detecting synchronization in spatially extended systems by complexity measurements. *Discrete Dynamics in Nature and Society* **9** (2005) 337-342
5. Toffoli, T., Margolus, N.: *Cellular automata machines: a new environment for modeling* (1987) MIT-Press
6. Ilachinski, A.: *Cellular automata: a discrete universe* (2001) World Scientific

On the Decidability of the Evolution of the Fuzzy Cellular Automaton 184

Angelo B. Mingarelli¹ and Samira El Yacoubi²

¹ Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6

² Laboratory of Mathematics and Physics for Systems (MEPS), University of Perpignan, 52, Paul Alduy Avenue 66860 - Perpignan Cedex. France

Abstract. In the previous paper [1] we presented general methods for detecting the evolution and dynamics of any one of the 255 fuzzy cellular automata (FCA) and showed that the method was applicable to all but nine of the 255 FCA. The main result there was that the limiting behavior of these FCA is decidable, except possibly for these nine, for finite initial configurations in a homogeneous background of zeros. Only six of these nine so called *exceptional* CA namely, FCA 172, 184, 202, 216, 226, and 228, appear to be interesting enough to warrant separate study, the other three, namely FCA 204, 228, and 240 being trivial. In this paper we study the exceptional FCA 184, a cellular automaton that admits a continuum of fixed points, namely the interval $[0, 1]$. This FCA is of interest because the general technique developed in [1] fails for the determination of its asymptotics. We show, in particular, that the asymptotic evolution of FCA 184 from any finite initial including random configuration of non-zero cells is decidable.

1 Introduction

The elementary cellular automata (ECA), considered by Wolfram and others, are wonderful examples of systems with simple rules that may produce unusually complex behavior; some of them, which are hardly predictable, are even capable of universal computation, [2]. On the other hand, one may get a better sense of their behavior by approximating their dynamics by continuous state CA's (or coupled map lattices), to wit, by "fuzzifying" the disjunctive normal form (the ECA is then called an FCA) and then studying the dynamics of the resulting global function as was done in [3]. Recent work in this new area of continuous CA's includes some variations on the game of Life in [4] and applications to pattern recognition [5]. In addition, such CA have been used to investigate the result of perturbations, for example, noisy sources, computation errors, mutations, etc. on the evolution of boolean cellular automata (cf., [6], [3], [7], etc.).

We recall some terminology for ease of exposition: A one-dimensional (boolean or binary-state) *cellular automaton* is a collection of ones and zeros (called an *initial string*) arranged on an infinite one-dimensional strip (where the space variable goes in the horizontal direction). Each cell is preceded by a cell behind it and another in front of it; these are called the *neighbors* of the cell. In the notation

of [8] these are known as “ $k = 2, r = 1$ CA’s”. We choose an arbitrary (but fixed) special element of this initial strip (at time $t = 0$) by labeling it as x_0 (or by x_0^0). Denoting the two-point set (or boolean space) consisting of the numbers 0 and 1 by $\{0, 1\}$ we define a *local rule* (sometimes also called a transition rule) by a function $g : \{0, 1\}^3 \rightarrow \{0, 1\}$, on consecutive triples x_{i-1}, x_i, x_{i+1} of this strip with the effect that $f(x)_i = g(x_{i-1}, x_i, x_{i+1})$ where the value $f(x)_i$ is placed directly below x_i , where time goes vertically down the diagram (on another such parallel strip which determines the states of the automaton at $t = 1$) and this update of the original cell values is performed for all the quantities in the initial string resulting in two parallel strips. The procedure is then repeated again and again to generate an infinite sequence of strips whose evolution at infinity we wish to study. The resulting diagram is called a light-cone or space-time diagram, see [8]. It follows that for a given general CA with transition rule g its states are related by

$$x_i^{t+1} = g(x_{i-1}^t, x_i^t, x_{i+1}^t), \quad t \geq 0.$$

Now, in the normal boolean case, each of three consecutive cells can have two values, namely, 0 and 1. Thus, the automaton is uniquely determined once its values on the eight basic elements 000, 001, 010, 100, 110, 101, 011, 111 are known. So there can be at most $2^8 = 256$ such CA’s and these are commonly known as the elementary cellular automata (ECA for short). To distinguish each automaton from every other in this list of several hundred, Wolfram [9] used a numbering scheme based on the binary expansion of each number n in the form $n = \sum_{i=0}^{\infty} r_i 2^i$, where $0 \leq n \leq 255$ and combined this with the disjunctive normal form. Specifically, every transition rule of such elementary CA’s is expressible as a *disjunctive normal form* (abbr. DNF), e.g.,

$$g(x_1, x_2, x_3) = \bigvee_{i|r_i=1} \bigwedge_{j=1}^3 x_j^{d_{ij}}$$

where d_{ij} is the j -th digit, from left to right, of the binary representation of i , and x^0 (resp. x^1) stands for $\neg x$ (resp. x). This DNF is simply a comprehensive device for the writing of any one of the 256 ECA’s. We are particularly interested in ECA 184 and its “fuzzification” defined below. Indeed, since $184 = 2^3 + 2^4 + 2^5 + 2^7$ we see that the so called rule number $184 = \sum_{i=0}^7 r_i 2^i$ forces $r_i = 1$ only for $i = 3, 4, 5, 7$. Use of the disjunctive normal form expression above gives us

$$\begin{aligned} g_{184}(x_1, x_2, x_3) &= \bigvee_{i|r_i=1} \bigwedge_{j=1}^3 x_j^{d_{ij}}, \\ &= (x_1^{d_{31}} \wedge x_2^{d_{32}} \wedge x_3^{d_{33}}) \vee (x_1^{d_{41}} \wedge x_2^{d_{42}} \wedge x_3^{d_{43}}) \vee (x_1^{d_{51}} \wedge x_2^{d_{52}} \wedge x_3^{d_{53}}) \\ &\quad \vee (x_1^{d_{71}} \wedge x_2^{d_{72}} \wedge x_3^{d_{73}}), \\ &= (x_1^0 \wedge x_2^1 \wedge x_3^1) \vee (x_1^1 \wedge x_2^0 \wedge x_3^0) \vee (x_1^1 \wedge x_2^0 \wedge x_3^1) \\ &\quad \vee (x_1^1 \wedge x_2^1 \wedge x_3^1), \\ &= (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee \\ &\quad \vee (x_1 \wedge x_2 \wedge x_3). \end{aligned} \tag{1}$$

The last equation represents the canonical expression of ECA 184.

It was pointed out in [3] that although ECA 184 belongs to (Wolfram’s) Class I and so is considered to be one of the simplest, its “. . . evolution in fuzzy backgrounds indicates otherwise . . . in particular the interaction between fuzziness and boolean cells creates complex structures. Even though it is difficult to theoretically explain this behavior, the experimental results quite clearly indicate that this is not a simple rule” [[3], p. 112].

A CA on $[0, 1]$ is said to be a *continuous CA* if its domain is $[0, 1] \times [0, 1] \times [0, 1] \equiv [0, 1]^3$ and its range is contained in the interval $[0, 1]$. If g is a general continuous CA on $[0, 1]$ its *fixed point set* is by definition the collection of all points $x \in [0, 1]$ such that $g(x, x, x) = x$. This set may consist of either a single point, a finite number of such points, or even the whole interval $[0, 1]$. The latter occurs for the special CA dubbed FCA 184 that we study in this paper. We now turn to the concept of fuzzy CA’s or, for brevity, FCA.

A *Fuzzy CA* is obtained by *fuzzification* of the local function of a Boolean CA: in the disjunctive normal form by redefining $(a \vee b)$ as $(a + b)$, $(a \wedge b)$ as (ab) , and $(\neg a)$ as $(1 - a)$. The usual fuzzification of the expression $a \vee b$ is $\max\{1, a + b\}$ so as to ensure that the result is not larger than 1. Note, however, that taking $(a + b)$ for the CA fuzzification does not lead to values greater than 1 since the sum of all the expressions for rule 255 is 1 (*i.e.*, $g_{255}(x, y, z) = 1$), and so every (necessarily non-negative) partial sum must be bounded by 1. Since every fuzzy rule is obtained by adding one or more of these partial sums it follows that every fuzzy rule is bounded below by 0 and above by 1. We refer the reader to the many articles in the references provided below, e.g. [1] for a brief explanation of the basic principles behind the notion of fuzzy CA, or FCA as we call them here. Recall that in “fuzzifying” the DNF above (1), we replace $\neg x$ by $1 - x$, $x \vee y$ by $x + y$, and $x \wedge y$ in (1) by their product, xy . By definition fuzzy CA are special cases of continuous CA.

Example 1. Referring to (1) and using the fuzzification procedure outlined above, we can write the form of FCA 184: Thus, starting from the disjunctive normal form of ECA 184,

$$\begin{aligned}
 g_{184}(x_1, x_2, x_3) &= (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee \\
 &\quad \vee (x_1 \wedge x_2 \wedge x_3) \\
 &\quad \text{we fuzzify it to find} \\
 &= (1 - x_1)x_2x_3 + x_1(1 - x_2)(1 - x_3) + x_1(1 - x_2)x_3 + x_1x_2x_3, \\
 &= x_1 - x_1x_2 + x_2x_3. \tag{2}
 \end{aligned}$$

The transition rule derived from (2) for $(x_1, x_2, x_3) \in [0, 1]^3$ in accordance with the fuzzification process defined here is denoted by FCA 184 (its two-state counterpart being given by (1)).

We suggest that one of the reasons that this FCA deserves special attention is that its fixed-point set is a continuum, namely the whole closed interval $[0, 1]$, (one of only 9 out of 255 such FCA, see [1] for the general theory). In addition, our purpose here is an attempt to explain previous insights in [3], at least theoretically. Generally speaking, although there are some exceptions, the evolution

of a space-time diagram is undecidable, at least in the special case of boolean ECA. In this note we will actually determine the evolution of the space-time diagram of this FCA thereby answering an open question in [3], and thereby showing that its evolution is decidable in sharp contrast with the boolean case.

Table 1. Partial space-time diagram of FCA 184 for a two-cell initial string with $x_0^0 = 0.95, x_1^0 = 0.34$, in a homogeneous background of zeros. We are interested in what happens as we proceed down the diagram?

0	.95	.34	0	0	0	0	0	0	0
0	.3230	.6270	.34	0	0	0	0	0	0
0	.2025	.3337	.4138	.34	0	0	0	0	0
0	.0676	.2730	.3363	.2731	.34	0	0	0	0
0	.0184	.1409	.2731	.3373	.1803	.34	0	0	0
0	.0026	.0543	.1946	.2418	.3378	.1190	.34	0	0
0	.0001	.0130	.0908	.2292	.2003	.3380	.0785	.34	0
0	.0000	.0013	.0327	.1159	.2510	.1591	.3382	.0518	.34

2 The Space-Time Diagram of FCA 184 and its Evolution over Time

Writing the canonical form of rule 184 as $g_{184}(x, y, z) = x - xy + yz$, its *diagonal function* $d(x) \equiv g_{184}(x, x, x)$ is given by solving the equation $d(x) = x$. Thus, every real number in $[0, 1]$ is a fixed point (it is *this* fact that characterizes the other eight exceptional rules as well).

Next, for a single seed a in a zero background observe that, by induction, $x_n^n = a$ for each $n \geq 1$, so that this rule is a right-shift. Clearly, for a single seed its dynamics are trivial. The difficulty occurs when we pass to the case of finite support/random initial strings of length greater than one.

Consider the case of two seeds, $a, b \in (0, 1)$, in a zero background. We take it that $x_0^0 = a, x_1^0 = b$ (so that x_1^0 is to the right of x_0^0). Recall [1] that the symbol $x_{\pm i}^j$ denotes the cell value i cells to the right (or left) of x_0^0 at time j (i.e, the j^{th} -row). The convergence of the right-diagonals and the dynamics of our FCA along them is of special interest in this article. It is easy to see that $x_n^{n-1} = b$ for all $n \geq 1$, so that the limit of this sequence (or zeroth diagonal, S_0^+), is $L_0^+ = b$. Next, we note that the terms of the first right-diagonal, S_1^+ , are given by $x_n^n = a(1 - b)^n$, a result that is easily verified by an induction argument. It follows that its limit, $L_1^+ = 0$, except in the special case where $b = 0$, in which case this reduces to the single seed scenario already discussed above. Difficulties arise in the discussion bearing on the next diagonal, S_2^+ . Applying the technique presented in [1] to this situation, one that was successful in [10] and [11], we have that $L_2^+ = g_{184}(L_2^+, L_1^+, L_0^+) = g_{184}(L_2^+, 0, b) = L_2^+$. Thus, no *a priori* information regarding L_2^+ is obtainable using the methods in [1], as they stand (and this is characteristic of these few exceptional FCA). Thus, a new technique

is needed and the following method serves as a blueprint that also applies to the remaining exceptional FCA. Observe that by definition,

$$x_n^{n+1} = x_{n-1}^n(1 - a(1 - b)^n) + ab(1 - b)^n,$$

for $n \geq 1$. Defining $aby_n \equiv x_n^{n+1}$ we get a simple recurrence relation for y_n , i.e.,

$$y_n = y_{n-1}(1 - a(1 - b)^n) + (1 - b)^n, \quad y_0 = 1,$$

where the solution, in this two-seed case, is given by

$$y_k = \prod_{j=0}^{k-1} (1 - a(1 - b)^{j+1}) \left\{ 1 + \sum_{i=0}^{k-1} \frac{(1 - b)^{i+1}}{\prod_{j=0}^i (1 - a(1 - b)^{j+1})} \right\} \quad (3)$$

for $k \geq 1$. Furthermore, since $|y_{n+1} - y_n| \leq (1 - b)^{n+1}$ it follows that the sequence $\{y_n\}$ is a Cauchy sequence. Its limit L_2^+ , found by passing to the limit as $k \rightarrow \infty$ in (3) and rearranging terms, is given by

$$L_2^+ = \prod_{j=0}^{\infty} (1 - a(1 - b)^{j+1}) + \sum_{i=0}^{\infty} (1 - b)^{i+1} \prod_{j=i+1}^{\infty} (1 - a(1 - b)^{j+1}), \quad (4)$$

where the infinite products are convergent by virtue of their form. Note that $L_2^+ \neq 0$ for $a, b \in (0, 1)$. The continuity of the transition rule for FCA 184 now implies that

$$L_3^+ = g_{184}(L_3^+, L_2^+, L_1^+)$$

a relation that forces $L_3^+ = 0$. The existence of L_4^+ is found by appealing to a recurrence relation once again but this time of the form $z_n \equiv x_n^{n+2}$,

$$z_n = z_{n-1}(1 - aby_n) + a^2y_nb(1 - b)^n,$$

whose solution can be written once again but is omitted here for brevity. Continuing in this way one can show by induction that $L_{2k-1}^+ = 0$, $L_{2k-2}^+ \neq 0$, for $k \geq 1$, found by solving first order recurrence relations leading to solutions involving complicated infinite products in the variables a, b . Clearly, this method leads to an algorithm for producing the limits L_k^+ of the right-diagonals of the space-time diagram of FCA 184. Note that the left diagonals produce zero limits at infinity.

The case of three or more seeds is similar to the two-seed case and is sketched for completeness. We assume a homogeneous background of zeros in which there is embedded the initial string $\dots 0, 0, 0, x_{-m}^0, \dots, x_0^0, x_1^0, 0, 0, 0, \dots$ of non-zero cell values. Then, as before, $L_0^+ = x_1^0$ and $x_n^n = x_0^0(1 - x_1^0)^n$ so that, once again, $L_1^+ = 0$. Taking $x_0^0x_1^0y_n \equiv x_n^{n+1}$ we get a corresponding recurrence relation for y_n ,

$$y_n = y_{n-1}(1 - x_0^0(1 - x_1^0)^n) + (1 - x_1^0)^n, \quad y_0 = 1.$$

The solution of the preceding equation is representable as a quantity analogous to (3) above so that the existence of L_2^+ is guaranteed and is an expression

similar to (4) involving infinite products. As in the two-seed case, the quantities $L_{2k-1}^+ = 0$, $L_{2k-2}^+ \neq 0$, for $k \geq 1$. This completes the analysis of the finite initial configuration case.

As for vertical columns/sequences of the form x_0^j, x_1^j , etc. for $j \geq 1$, it is unlikely that these vertical sequences will converge at all due to the alternating nature of zero limits (i.e., $L_{2k-1}^+ = 0$) along right-diagonals. Specifically, if one can show that the limit of the limits, $L_{2k-2}^+ \rightarrow 0$, as $k \rightarrow \infty$ (which we cannot prove) then one can infer that the vertical sequences will all converge to zero. Numerical evidence though seems to support this claim (cf., Table 1). On the other hand, if

$$\liminf_{k \rightarrow \infty} L_{2k-2}^+ > 0,$$

then for each $i \geq 0$, $\lim_{k \rightarrow \infty} x_i^k$ does not exist.

3 Conclusion

We have shown that FCA 184, one of six exceptional fuzzy cellular automata [1] whose set of fixed points is a continuum, has complicated behavior at infinity but does not lend itself to complexity, thus answering a question raised in [3]. Indeed, the asymptotic nature of its space-time diagram is deterministic and apparently non-chaotic, the evolution from any finite random initial string of fuzzy values in the open unit interval being decidable, as opposed to its boolean counterpart.

Acknowledgments

This research is partially supported by an NSERC Canada Research Grant to the first author.

References

1. Angelo B. Mingarelli, *The global evolution of general fuzzy cellular automata*, Int. J. Unconventional Comp., Vol. X, (2005), 1-24.
2. S. Wolfram, *A New Kind of Science*, Wolfram Media, Champaign, IL, 2002.
3. G. Cattaneo, P. Flocchini, G. Mauri, C. Quaranta Vogliotti, N. Santoro, *Cellular automata in fuzzy backgrounds*, Physica D **105** (1997), 105-120
4. C. A. Reiter, *Fuzzy automata and life*, Complexity, **7** (3) (2002), 19-29.
5. P. Maji, R. Nandi, P. Chaudhuri, *Design of fuzzy cellular automata (FCA) based pattern classifier*, in Fifth International Conference on Advances in Pattern Recognition, ICAPR-2003, December 10-13, Calcutta, India. To appear.
6. G. Cattaneo, P. Flocchini, G. Mauri, and N. Santoro, *Fuzzy cellular automata and their chaotic behavior*, in Proc. International Symposium on Nonlinear Theory and its Applications, Hawaii, IEICE Volume 4, (1993) 1285-1289
7. P. Flocchini, F. Geurts, A. Mingarelli, N. Santoro, *Convergence and aperiodicity in fuzzy cellular automata: revisiting rule 90*, Physica D, **42** (2000), 20-28.

8. T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, T. Pellizzari, *Computation in cellular automata: A selected review* in Nonstandard Computation: Weinheim: VCH Verlagsgesellschaft (1998), 95-140.
9. S. Wolfram, *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.
10. Angelo B. Mingarelli, *Fuzzy rule 110 dynamics and the golden number*, WSEAS Trans. Computers, 2 (4) (2003), 1102-1107.
11. Angelo B. Mingarelli and Elzbieta Beres, *The dynamics of fuzzy cellular automata: Rule 30*, WSEAS Trans. Circuits and Systems, (10) **3**, 2211-2216.

Cell Dormancy in Cellular Automata

Mohammad Ali Javaheri Javid and Rene te Boekhorst

School of Computer Science
Faculty of Engineering and Information Sciences
University of Hertfordshire
Hatfield, Hertfordshire AL10 9AB
United Kingdom
javaheri@javaheri.info, r.teboekhorst@herts.ac.uk

Abstract. This paper describes a novel implementation of a two-dimensional Cellular Automaton (CA) by introducing a *dormant* state. An overview of the use of CA's in the field of Artificial Life reveals that certain crucial aspects of biological realism have been sacrificed in favour of abstraction or have not been considered at all. Conway's famous "Game of Life" model includes certain fundamental aspects of population dynamics, including the transition from living state to dead state. But even the simplest biological system consists of more stages than the binary state in the Game of Life. The aim of this research is to build an extended CA model of natural biological systems by introducing a dormant state and to investigate the effect of dormancy on simple population dynamics.

1 Introduction

The rapidly growing field of Artificial Life is inspired by biological systems and exploits computer technology to synthesize and simulate common processes and characteristics of living organisms. Artificial Life can be used as powerful tool in a variety of fields like psychology, biology, therapy, robotics, Artificial Intelligence, evolutionary computation, art, etc [1]. An overview of the field reveals that some aspects of biological systems lost in abstraction or simply have not been considered. The famous Game of Life introduced by John Conway resembles the population dynamics by including the transition from living state to dead state. However, observations from developing real biological systems show that they go through more stages than the binary states of the Game of Life. In this paper we consider a more realistic CA model of simple biological systems in the theoretical framework of Artificial Life by including and investigating the effect of dormancy on the dynamics of an artificial population.

2 Background

Artificial Life considers living systems as dynamical systems in which life is an emergent rather than an inherent property of those systems. This view departs from

the classical biological framework which studies living systems by decomposing them from top to down in order to understand the mechanism of life. The top-down approach describes the mechanics of life in terms of complex, hierarchical biochemical system. However, life cannot be understood completely in terms of its parts, Instead, the exhibited properties of living systems are the result of interactions between their components. Artificial Life, by adapting a bottom-up approach, tries to synthesize living systems by combining separate elements or substances to form a coherent whole [2] in a medium like computers or robots. In this approach, the whole is different than the sum of its constituents.

2.1 Cellular Automaton and the Game of Life

Cellular Automata (CA) are dynamical systems in which both space and time are represented as discrete units. They have been conceived by von Neumann and Ulam to provide a formal framework for investigating the behaviour of complex, extended systems [3] and to study the process of self-reproduction. Von Neumann was interested in the essence of reproduction and not in any particular, implementation of the process. Thus, he purposely abstracted away all the details of, say, how animals reproduce, and instead concentrated on the simplest mathematical framework that allows information to reproduce [4].

A cellular automaton consists of an array of cells, each of which can be in one of a finite number of possible states (known as *finite-state automaton*), updated synchronously or asynchronously [see 5] in discrete time steps, according to local interaction rules. The state of a cell at the next time step is determined by the current states of a surrounding neighbourhood of cells [6, 7].

Conway introduced the “Game of Life” as a very simple Boolean CA [8]. But despite its simplicity, the “Game of Life” gives rise to complicated behaviour. The cells in the Game of Life have two states, $\{0, 1\}$, which are interpreted as *live* or *dead* states.

The rule structure of CA in the Game of Life is given a Moore Neighbourhood (eight possible neighbours: N, W, S, E, NE, NW, SE and SW of a reference cell):

1. A cell becomes alive if three of its neighbours are alive.
2. A cell remains alive if two or three of its neighbours are alive, otherwise it dies either from exposure (number of neighbours ≤ 1) or from overcrowding (number of neighbours ≥ 4).

3 Life and Death in Biological and Synthesised Systems

Any attempt in creating new forms of life in artificial media must have clear idea about the nature of life. Life can be studied at a molecular,- cellular,- organismal-, population - and ecosystem level [9]. Next, one could try to extract key properties of life which occur in all the stated levels. So far, attempts to define life are based on listing such properties. Properties common to many lists include the abilities to replicate, evolve, metabolize, respond to stimuli, and to repair damage [10].

Problem arises from a lack of agreement on what should or should not be included in the list [see 11, 12]. The properties that are considered to distinguish a living system from a non-living one are not consistent enough to give an appropriate overview of living systems. This raises the problem that if a system lacks the properties of life, then the system could be either dead or “just not alive”. According to our knowledge, now non-living system are not distinguished from dead systems in the literature so far.

4 Dormancy

Dormancy is a well-known feature of many biological systems. Biological systems (e.g. bacteria) due to stress conditions (say lack of nutrition) may enter into dormant state with relatively low metabolism and ceased replication. As the environmental condition changes, the system regains its active state properties like high metabolism and reproduction. Dormancy is considered to have survival value in that it helps to endure stressful circumstances. This metabolically inactive but reversible state may occur at different levels. When it involves the whole organism, it is more likely to be periodic like winter hibernation in animals and seasonal dormancy in plants. In some cases dormancy is limited to reproductive organs like seeds and spores. In all cases, the system regains its active state properties once the environmental conditions have become favourable again.

Stress conditions to which many bacteria are being exposed apparently did not call for the evolution of highly sophisticated resistant structures [13]. Bacteria in the dormant state function at very low metabolic rates and do not undergo cell division. When the stress is released, bacteria resume cell division. Thus, the transformation to a dormant state considered here involves adaptive mechanisms but no morphological differentiations such as sporulation [14, 15].

5 Dormancy in Artificial Systems

Reproduction is a fundamental character of living systems. In von Neumann's initial thought-experiment, artificial reproduction was realized in a universal constructor surrounded by components: given the description of any machine, it will locate the proper parts and re-construct that machine. If given a description of itself, it will construct itself [2]. Thus, reproduction can be viewed as self-reconstruction, explained in terms of interactions of simple elements and studied as logical principles independent of its physical realization [16].

Von Neumann's model and Langton's loops assume certain minimal requirements for a system to exhibit self-reconstruction. However, should a system be called “dead” or “not alive” when these requirements are not met? In this paper, and inspired by the existence of dormant stages in biological systems, we would like to extend these notions by introducing an intermediate state in artificial systems. For this purpose, we define a dormant state as a state in which a system fails to exhibit the properties of living systems. However it is able to reverse to a previous state in which

it is able to display these properties. Translating this definition in the context of a two-dimensional CA given a Moore Neighbourhood, a dormant cell is a cell which in a previous time step ($t-1$) was in the living state $\{1\}$ and in the time (t) it is in the dormant state $\{0.5\}$. Considering the state transition of dormant cells in biological systems, we distinguish the following types of dormancy for cells in a CA:

Dormancy A: This is a static dormancy which maintains its dormant value of $\{0.5\}$ for all time steps.

1. A cell becomes alive if three of its neighbours are alive (reproduction).
2. A cell remains alive if two or three of its neighbours are alive otherwise it enters into a dormant state.

These rules are identical to Conway's Game of Life, the only difference being that cells enter into dormant state (and therefore still occupy space) instead of dying (and hence leave behind an open space).

Dormancy B: This is a type of dormancy which, due to ongoing environmental stress can change a dormant state $\{0.5\}$ into a dead state of $\{0\}$:


1. An empty cell becomes alive if three of its neighbours are alive (reproduction), otherwise it stays empty
2. A cell stays alive if two or three of its neighbours are alive otherwise it enters into the dormant state (maintenance)
3. A cell remains dormant if less than four of its neighbours are dormant otherwise it dies.

Dormancy C: This includes a phase of dormancy which, due to favourable environmental factors, regains the state of being alive $\{1\}$:

1. A cell becomes alive if three of its neighbours are alive, otherwise it stays empty.
2. A cell remains alive if two or three of its neighbours are alive otherwise it enters into dormant state.
3. A cell remains dormant if four of its neighbours are dormant and dies if it has more than 4 neighbours.
4. A dormant cell becomes alive if two or three of its neighbours are dormant.

The rule number four retains the situation that due to environmental changes which favour reproduction, dormant cells will be able to regain their live states.

6 Simulations and Results

The rule tables of the three systems were implemented in a CA environment with wrap-around Moore neighbourhood to test the effect of a set of initial conditions. The experimental initial states are the "glider" and a rectangular "block" with a perimeter of 14 cells (). Figure 1 illustrates snapshots from some later stages in the pattern development for dormancy type A starting from a "glider" and figure 2 shows the same when starting with a "block". Figures 3 to 6 show the results of runs for both

initial configurations for respectively dormancy types B and C. In figures (1 to 6) the final state is a state that the growth of cells continues unless there is no place to grow (Full colour snapshots are available from:<http://www.javaheri.info/dormancy.html>).

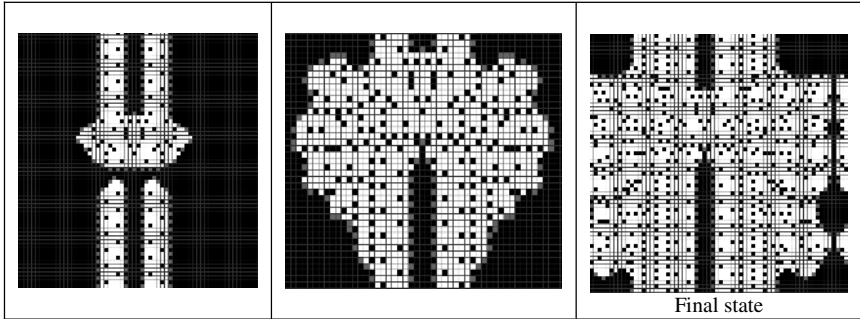


Fig. 1. Development of patterns in dormancy type A for Glider initial state. White = dormant cells, Grey = living cells.

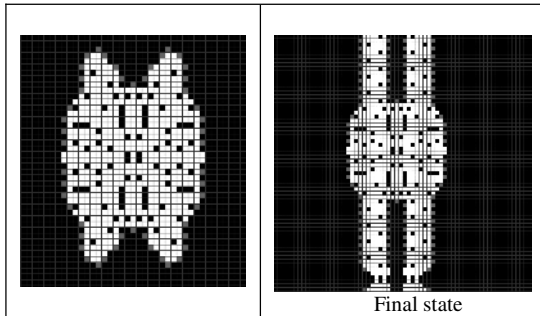


Fig. 2. Development of patterns in dormancy type A for Block initial state

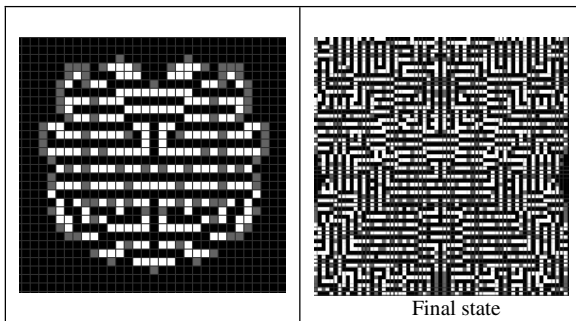


Fig. 3. Development of patterns in dormancy type B for Glider initial state

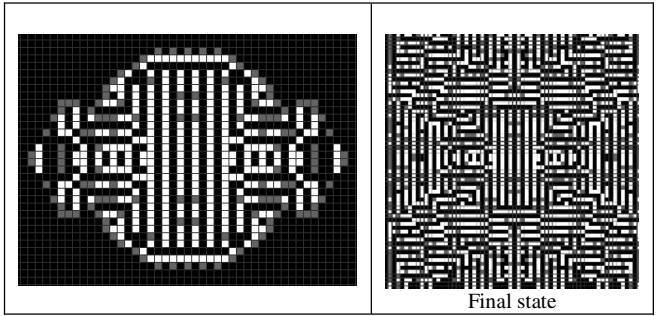


Fig. 4. Development of patterns in dormancy type B for Block initial state

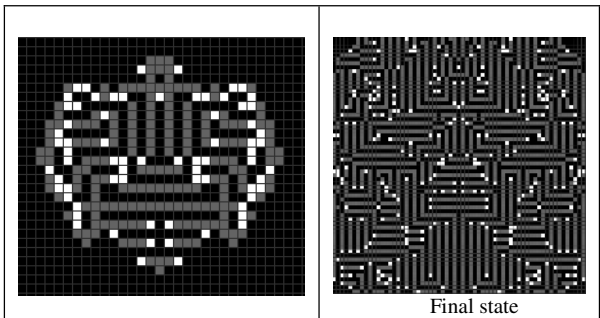


Fig. 5. Development of patterns in dormancy type C for Glider initial state

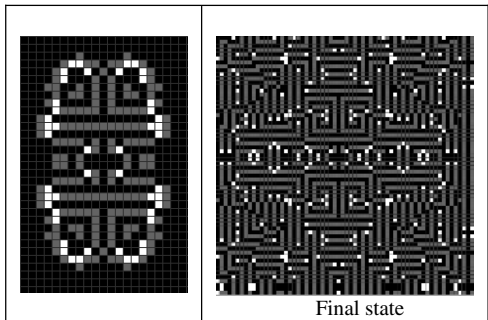


Fig. 6. Development of patterns in dormancy type C for Block initial state

The experiment conducted for ten consecutive time steps and compared with those of the classical Game of Life. Simulations show that from the third time step the Game of Life results in a fixed number of living cells, i.e. an attractor. The development of the systems including dormancy (A, B and C) is very different. Obviously, dormancy affects the population dynamics of artificial cells in a CA environment. Figure 7 shows that the number of living cells is increased in systems that include dormant states, which suggests survival value.

The initial state plays an important role. If living cells are in close proximity of each other - even if randomly located - their growth continues unless there is no place to grow (final state). We found that for dormancy type A, cell growth occurs at the edges of an area covered by dormant cells (figure 1 and 2). Furthermore, in dormancy type A for Glider initial state, after time step {10}, the population splits into two parts and only grows upwards (figure 1) which is reminiscent of a kind of directed tissue growth. In a torpid CA environment, when the two parts meet each other, there is explosion of population growth which accelerates the growth of dormant cells. We also noticed that the final steady state for both Glider and Block initial states are symmetric. As the figures 1 - 6 show, the final state of all dormancy types produce a symmetric pattern.

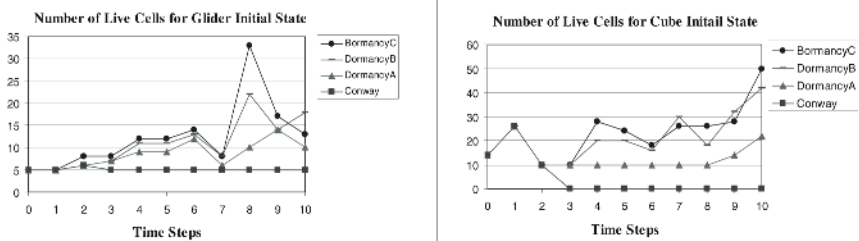


Fig. 7. Dynamics of the number of living cells in different dormancy systems compared to Conway's Game of Life

7 Conclusions and Further Work

The Game of Life is a very simple two-dimensional CA which abstracts natural biological systems into two discrete states. Inspired by natural biological systems we implemented a Game of Life with an intermediate dormant state. This led to the development of a model that describes different types. Analysis of the runs showed that the introduction of dormant state significantly affects the population dynamics of living cells. The increased number of living cells in different dormancies suggests that dormancy has survival value for an artificial system in a CA environment. The shape of the patterns depends on the initial configuration and their growth pattern is symmetric. As possible further work for our dormancy models, we suggest investigating whether there is relationship between number of dormant cells and living cells.

References

1. Langton C. G. and Shimohara K.: *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living systems*, The MIT Press/A Bradford Book (1996).
2. Langton, C. G.: *Artificial Life*, In Langton C.G. (ed.) *Artificial Life*, Addison-Wesley, (1989) 1- 47.
3. von Neumann, J.: *Theory of Self-Reproducing Automata*, University of Illinois Press, Illinois. Edited and completed by A. W. Burks (1996).

4. Flake, G.W.: The computational beauty of Nature, MIT press (1998).
5. Nehaniv, C. L.: Evolution in asynchronous cellular automata, Proceedings of the eighth international conference on artificial life, (2002) 65-73.
6. Wolfram, S.: Universality and complexity in cellular automata. *Physica D*, Vol. 10 (1984)1-35.
7. Tofoli, T. and Margolus, N.: *Cellular Automata Machines*, The MIT Press, Cambridge, Massachusetts (1987)
8. Gardner, M.: The fantastic combinations of John Conway's new solitaire game "Life". *Sci. Am.* 223 (1970) 120-123.
9. Taylor Charles and Jefferson David: Artificial life as a tool for biological inquiry. In Christopher G. Langton (ed), *Artificial Life: an overview*, Cambridge, Massachusetts, 1995. MIT Press (1995) 1-14.
10. Ray, T. S.: Artificial Life, In Renato Dulbecco, David Baltimore, François Jacob, Rita Levi-Montalcini (eds), *Frontiers of Life*, Vol. 1, One The Origins of Life, Academic Press (2001) 107-124.
11. Emmeche, C.: *The Garden in the Machine*, Princeton (1994).
12. Adami, C.: *Introduction to Artificial Life*. Springer-Verlag, Berlin (1998).
13. Heins, Y.: *Survival and Dormancy of Microorganism*, John Wiley & Sons (1987).
14. Roszak, D. B., and R. R. Colwell.: Survival strategies of bacteria in the natural environments, *Am. J. Pub. Health*, Vol. 51 (1987) 365-379.
15. Barer, M.R. and Harwood, C.R. (1999) Bacterial viability and culturability, *Adv. Microb. Physiol.*, Vol. 41 (1999) 93 -137.
16. Heudin, J.C. (1998) *Virtual Worlds*, Proceedings of the First Int. Conf. on Virtual Worlds, Vol. 1434, Springer-Verlag Lecture Notes in Computer Science (Berlin) (1998).

Introduction to the ICCS2006 Workshop on Dynamic Data Driven Applications Systems

Frederica Darema

National Science Foundation, Arlington VA 22230, USA
darema@nsf.gov

Abstract. The Dynamic Data Driven Application Systems (DDDAS) concept entails the ability to incorporate dynamically data into an executing application simulation, and in reverse, the ability of applications to dynamically steer measurement processes. Such dynamic data inputs can be acquired in real-time on-line or they can be archival data. DDDAS offers the promise of improving modeling methods, augmenting the analysis and prediction capabilities of application simulations, improving the efficiency of simulations and the effectiveness of measurement systems.

The scope of the present workshop provides examples of research and technology advances in enabling the DDDAS capabilities.

1 Introduction

The Dynamic Data Driven Application Systems (DDDAS) [1,2] concept entails the ability to incorporate dynamically data into an executing application simulation, and in reverse, the ability of applications to dynamically steer measurement processes. Such dynamic data inputs can be acquired in real-time on-line or they can be archival data. The DDDAS concept offers the promise of improving modeling methods, augmenting the analysis and prediction capabilities of application simulations, improving the efficiency of simulations and the effectiveness of measurement systems. Advances and technology capabilities required and enabled through the DDDAS concept are discussed in [3 and 4], and are fostered through the DDDAS Program [5] announced in 2005, with seeding efforts in the area having started previously (2000 – 2005) through the NSF ITR Program [6]. The DDDAS Program was co-sponsored by multiple Directorates and Offices of NSF, NOAA and NIH, and in cooperation with Programs in the European Community and the United Kingdom [referenced in 5].

In [4 and 5] it is emphasized the need for synergistic multidisciplinary research in applications modeling, in mathematical and statistical algorithms, in measurement methods and data management, and in computer systems software that's necessary to enable DDDAS capabilities. Furthermore in [4] the case is made that the DDDAS concept requires and will push for infrastructure that goes beyond the traditional notions of computational grids and provide new directions and capabilities in CyberInfrastructure environments [7]. While fundamental research advances are imperative to enable DDDAS capabilities, and the impact of DDDAS to scientific and

engineering fields is becoming widely recognized, there is increasing acceptance of the DDDAS potential to many industrial environments and such opportunities are manifested through the participation of industry in a number of the research projects.

The efforts presented in this workshop provide examples of technologies and capabilities that are being developed based on the DDDAS concept and supporting DDDAS environments. In www.cise.nsf.gov/dddas, a listing of all projects funded under the DDDAS rubric can be found; the papers presented here represent a sample of the range of projects and scope of the technical areas pursued. This introduction provides an overview and the context of the work presented. The present workshop is the fourth of a series of workshops on the DDDAS topic that have been organized as part of ICCS, starting with ICCS'03. Other community outreach activities include the NSF sponsored workshops [1 and 8].

2 Overview of Work Presented in This Workshop

The papers presented in this workshop represent ongoing research projects each addressing a number of aspects along the technical opportunities derived from the DDDAS concept as well as the challenges and technical advances required to enable DDDAS capabilities. With the exception of projects in [21] and [32-35], all other projects are enabled with the support of the ITR and DDDAS programs referenced above.

Several of the papers presented address application simulations whose capabilities are enhanced by dynamic data inputs into the executing application/simulation. Many of the papers presented involve multidisciplinary work to develop new capabilities along the four components: applications and algorithms, together with advances in computer systems software and measurement methods. While the emphasis of each project on each of the four components may vary, in most of these papers the application advances and enablement are made in synergism with in the underlying software technology, supporting the development of the complex DDDAS applications, their dynamic runtime support requirements and interfaces to control measurement systems. Several papers provide examples of simulation controlled measurement processes (instruments and sensor networks). Most of the work on mathematical algorithms tolerant and stable to dynamically streamed data is done in the context and together with the development of the application models. In all cases the mathematical algorithms and systems software development is done in the context of specific applications driving and validating the developed technologies.

The projects discussed in [9-19 and 22-30] enable improved analysis and prediction capabilities of application simulations and/or speeding up the computation times together with improved quality of simulations, and in a feedback loop the simulations control of measurements, sensors and other instruments.

In projects [9-12], the applications are in the areas of environmental and natural resource management. In [9] the project employs previous work performed by the investigators in the Instrumented Oil-Field DDDAS project that has enabled a new generation of data-driven, interactive and dynamically adaptive strategies for subsurface characterization and oil reservoir management. This previous work has led to the implementation of advanced multi-physics, multi-scale, and multi-block

numerical models and an autonomic software stack for DDDAS applications. The software stack implements a Grid-based adaptive execution engine, distributed data management services for real-time data access, exploration, and coupling, and self-managing middleware services for seamless discovery and composition of components, services, and data on the Grid. This paper investigates how these solutions can be leveraged and applied to address another DDDAS application of strategic importance, e.g. the data-driven management of Ruby Gulch Waste Repository. Work conducted in [10] develops capabilities to convert a traditional data collection sensor and ocean model into a DDDAS enabled system for identifying contaminants, dynamically coupling different models, simulations, and sensing strategies, in a symbiotic manner. The paper in [11] is centered on methods for detection, management and mitigation of threat in drinking water distribution systems; it involves real-time characterization of any contaminant source and plume, design of control strategies, and design of incremental data sampling schedules. This requires dynamic integration of time-varying measurements along with analytical modules that include simulation models, adaptive sampling procedures, and optimization methods. These modules are compute-intensive, requiring multi-level parallel processing via computer clusters. Since real-time responses are critical, the computational needs must also be adaptively matched with available resources. This requires a software system to facilitate this integration via a high-performance computing architecture such that the measurement system, the analytical modules and the computing resources can mutually adapt and steer each other. The paper describes the development of such an adaptive cyberinfrastructure system facilitated by a dynamic workflow design. The paper in [12] is aimed at understanding environmental change through the deployment of next-generation wireless sensor networks and assimilating heterogeneous data, assessing the relative value and costs of data collection, and scheduling activities accordingly. Thus, they are dynamic, data-driven distributed systems that integrate sensing with modeling and prediction in an adaptive framework. Integration of a range of technologies will allow estimation of the value of future data in terms of its contribution to understanding and cost. This balance is especially important for environmental data, where sampling intervals will range from meters to entire landscapes, and from seconds to years in duration. The paper describes a general framework for dynamic data-driven wireless network control that combines modeling of the sensor network and its embedding environment, both in and out of the network, and describes a range of challenges that must be addressed, and an integrated suite of solutions for the design of dynamic sensor networks.

The papers in [13-15] are in application areas dealing with security and crisis management and mitigation. The paper in [13] is developing a prototype emergency response system. This dynamic data driven application system (DDDAS) uses wireless call data, including call volume, who calls whom, call duration, services in use, and cell phone location information to utilize the cell phones of a city to serve as an ad hoc mobile sensor net, measuring the movement and calling patterns of the population. Social network theory and statistical analysis on normal call activity and call locations establish a baseline. A detection and alert system monitors streaming summary cell phone call data. Abnormal call patterns or population movements trigger a simulation and prediction system. The project in [14] is concerned with applying Dynamic Data Driven Application Simulations (DDDAS) to monitor and

manage surface transportation systems. Building upon activities such as the Vehicle-Infrastructure Integration initiative, a hierarchical DDDAS architecture will be created that includes coupled in-vehicle, roadside, and traffic management center simulations. The overall architecture and further advances incorporating the DDDAS concept will be deployed to implement and evaluate the effectiveness of the intended system for a portion the Atlanta area in the context of a hypothesized emergency evacuation scenario. The paper in [15] will apply DDDAS capabilities developed by the investigators that allow to study interaction between fire and agent models during a fire evacuation. The analysis from that research can be used to validate proposed ideas in evacuation and building designs to ensure safety of buildings given various agent behaviors. Through study of the interactions, a better understanding is gained of how to improve building design, evacuation strategies and regulations, as well as training first responders.

The papers in [16-17] are addressing application areas of critical infrastructure management and protection, such as electrical power systems management. The paper in [16] is focused in developing methods and approaches for maintaining acceptable electric transmission system reliability and optimized electric energy delivery, requires innovations in sensing, diagnostics, communications, data management, processing, algorithms, risk assessment, decision-making (for operations, maintenance, and planning), and process coordination. The work presented is a comprehensive approach to develop methods and processes in these areas, driven by the ultimate objective to develop a hardware-software prototype capable of auto-steering the information-decision cycles inherent to managing operations, maintenance, and planning of the high-voltage electric power transmission systems. The paper in [17] is also concerned with fault-tolerant operation of large-scale electric power systems through efficient and methods that employ the DDDAS concept to enable fast and robust reduced-order-modeling and filtering methods and merging of data from multiple sources.

The papers in [18-21] address methods for structural monitoring and fault-tolerance in structures like buildings and aircraft. Paper [18] discusses methods and implementation scope supporting a dynamic data driven application system for material and structural health monitoring as well as critical event prediction. The dynamic data driven paradigm is exploited to promote application advances, application measurement systems and methods, mathematical and statistical algorithms and systems software infrastructure relevant to this effort. These advances are intended to enable behavior monitoring and prediction as well as critical event avoidance on multiple time scales. Paper [19] uses DDDAS to enable structural sensing and control. Paper [20] is employing DDDAS for evaluation of fluid-thermal systems wherein a complete specification of the boundary conditions is not known a priori and experimental diagnostics are restricted to a limited region of the flow-field. The methodology is applied to the configuration of a heated jet injected into a laminar boundary layer where the jet temperature is not known a priori. Papers [21, 37] use DDDAS to improve methods for analysis, prediction of atmospheric contaminant propagation and inversion for identification of the origin of the contaminant, employing the concept to steer sensors and stream up-to-date information to enhance reduced order models and PDE solvers approaches, in a feed-back control loop.

Papers [22-26] address topics of atmospheric, aqueous and ambient environmental systems such as: In [22] a data assimilation technique for coupled ionospheric and thermospheric dynamics. Using a frozen linear dynamics matrix for the time update of the error covariance and the evaluation of the Kalman filter gain, and demonstrating the performance of the data assimilation technique on a section of the ionosphere. Paper [23] uses the DDDAS concept to simultaneously factor position and amplitude errors for improved prediction of path of rapidly changing mesoscale events like hurricanes. In [24], in the context of a marine biology application, observing systems facilitate scientific studies by instrumenting the real world and collecting corresponding measurements, with the aim of detecting and tracking phenomena of interest. This project focuses on a class of observing systems, which are embedded into the environment, consist of stationary and mobile sensors, and react to collected observations by reconfiguring the system and adapting which observations are collected next. Paper [26] reports on an ongoing effort to build a Dynamic Data Driven Application System (DDDAS) for short-range forecast of weather and wildfire behavior from real-time weather data, images, and sensor streams. The system changes the forecast as new data is received. The methods involve applying an ensemble Kalman filter in time-space with a highly parallel implementation, and demonstration of the validity of the derived system by using a DDDAS test-bed approach and data collected from an earlier fire.

Applications in [27-29] are in the areas of health and behavioral sciences. The project in [27] uses the DDDAS concept to enable a dynamic data-driven planning and control system for laser treatment of cancer. The research includes development of a general mathematical framework and a family of mathematical and computational models of bio-heat transfer, tissue damage, and tumor viability, dynamic calibration, verification and validation processes based on laboratory and clinical data and simulated response, and (3) design of effective thermo-therapeutic protocols using model predictions. At the core of the proposed systems is the adaptive-feedback control of mathematical and computational models based on a posteriori estimates of errors in key quantities of interest, and modern Magnetic Resonance Temperature Imaging (MRTI), and diode laser devices to monitor treatment of tumors in laboratory animals. This approach enables an automated systematic model selection process based on acceptance criteria determined a priori. The methodologies to be implemented involve uncertainty quantification methods designed to provide an innovative, data-driven, patient-specific approach to effective cancer treatment. Paper [28] presents a case of interdisciplinary collaboration in building and using a set of tools to compute the flows in a branched artery, to compare them with prior physical flow visualization, and to interpret them with further users in mind. The geometry is of a typical epicardial coronary artery with a side branch. The incompressible Navier-Stokes equations are solved with the hybrid spectral/hp element solver Nektar, and simulations were visualized in the CAVE, an immersive 3D stereo display environment, which facilitates interpretation of simulated features. Paper [29] describes a novel approach towards the development of new design architectures and research test-beds for advanced Brain Machine Interfaces (BMIs) to help paralyzed patients, and others with motor disabilities, regain (artificial) motor control and autonomy. It addresses a critical design challenge in deriving the functional mapping between the subject's movement intent and actuated

behavior. Inspired by motor control research, this paper considers a predictive framework for BMI using multiple adaptive models trained with supervised or reinforcement learning in a closed-loop architecture that requires real-time feedback. In this framework, BMIs require a computing infrastructure capable of selectively executing multiple models on the basis of signals received by and/or provided to the brain in real time. Middleware currently under investigation to provide this data-driven dynamic capability is discussed. Paper [30] presents a dynamic data-driven framework for tracking gestures and facial expressions from monocular sequences. The system uses two cameras, one for the face and one for the whole body view for processing in different scales. Specifically, and for the gesture tracking module, the system tracks the hands and the head, obtaining as output the blobs (ellipses) of the ROIs, and detects the shoulder positions with straight lines. For the facial expressions, the approach is to extract the 2D facial features, using a fusion between KLT tracker and a modified Active Shape Model, and then we obtain the 3D facemask with fitting a generic model to the extracted 2D features. The main advantages of the system pursued are: adaptivity, i.e. robustness to external conditions, such as lighting, and independent from the examined individual, and computational efficiency, providing us results off- and online with a rates higher than 20frames/sec.

In [31] the work presented employs artificial intelligence (AI) approaches to enable the management of a data driven simulation system, in particular with regard to adaptive selection of data and refinement of the model on which the simulation is based. The approaches used consider two different classes of intelligent agent that can control a data driven simulation: (a) an autonomous agent using internal simulation to test and refine a model of its environment and (b) an assistant agent managing a data-driven simulation to help humans understand a complex system (assisted model-building). In the first case the agent is situated in its environment and can use its own sensors to explore the data sources. In the second case, the agent has much less independent access to data and may have limited capability to refine the model on which the simulation is based. This is particularly true if the data contains subjective statements about the human view of the world, such as in the social sciences. For complex systems involving human actors, the paper pursues an architecture in which assistant agents cooperate with autonomous agents to build a more complete and reliable picture of the observed system.

The paper in [32] presents tools that serve to capture and apply a scientist's insight about opportunities for, and limitations of, simulation adaptation, and reports on the two ongoing collaborations that serve to guide and evaluate the methods pursued. One of the intended consequences of utilizing simulations in dynamic, data-driven application systems is that the simulations will adjust to new data as it arrives. These adjustments will be difficult because of the unpredictable nature of the world and because simulations are so carefully tuned to model specific operating conditions. Accommodating new data may require adapting or replacing numerical methods, simulation parameters, or the analytical scientific models from which the simulation is derived. This research project emphasizes the important role a scientist's insight can play in facilitating the runtime adaptation of a simulation to accurately utilize new data.

The papers [33-36] present work that can be evolved as enabling algorithmic and computer systems software technologies supporting DDDAS environments. In [33] web services, service oriented architecture and model driven architecture approaches

are discussed; in [34] advanced visualization methods, in [35] multivariate signal analysis methods are discussed, and in [36] knowledge-based methods for data management approaches are presented. The methods discussed in these papers are needed advances for capabilities such as dynamic composition of DDDAS applications and dynamic data management interfaces.

3 Summary

The DDDAS concept opens new capabilities in application and measurement methods and systems. Through efforts that have started developing DDDAS systems, such promise begins to materialize, and we are in a steady path towards accomplishing that objective. The progress made is exemplified by the projects presented here.

Acknowledgements

I'm indebted to all my colleagues at NSF, NIH and NOAA, as well as the cooperating EU-IST and e-Sciences programs, who have worked with me on the DDDAS Program Solicitation and in cosponsoring the funded projects.

References

1. NSF Workshop, March 2000; www.cise.nsf.gov/dddas
2. F. Darema, Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements, ICCS'04
3. F. Darema, Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems, Proceedings of the IEEE, Special Issue on Grid Computing, 3/05
4. F. Darema, Dynamic Data Driven Applications Systems: New Capabilities for Application Simulations and Measurements, ICCS'05
5. DDDAS-Dynamic Data Driven Applications Systems Program Solicitation (NSF 05-570); www.cise.nsf.gov/dddas
6. NSF Information Technology Research (ITR) Program (1999-2004)
7. Cyberinfrastructure Report http://www.communitytechnology.org/nsf_ci_report
8. NSF Sponsored Workshop: DDDAS-Dynamic Data Driven Applications Systems, Jan 19-20, 2006 [www.cise.nsf.gov/dddas]
9. Manish Parashar, Vincent Matossian, Hector Klie, Sunil G. Thomas, Mary F. Wheeler, Tahsin Kurc, Joel Saltz, and Roelof Versteeg; Towards Dynamic Data-Driven Management of the Ruby Gulch Waste Repository
10. Craig C. Douglas, J. Clay Harris, Mohamed Iskandarani, Chris R. Johnson, Robert Lodder, Steve Parker, Martin J. Cole, Richard Ewing, Yalchin Efendiev, Raytcho Lazarov, and Guan Qin; Dynamic Contaminant Identification in Water
11. Kumar Mahinthakumar, Gregor von Laszewski, Ranji Ranjithan, Downey Brill, Jim Uber, Ken Harrison, Sarat Sreepathi, and Emily Zechman; An Adaptive Cyberinfrastructure for Threat Management in Urban Water Distribution Systems:
12. Paul G. Flikkema, Pankaj K. Agarwal, James S. Clark, Carla Ellis, Alan Gelfand, Kamesh Munagala, and Jun Yang; Model-Driven Dynamic Control of Embedded Wireless Sensor Networks

13. Greg R. Madey, Gabor Szabo, and Albert-László Barabási; WIPER: The Integrated Wireless Phone Based Emergency Response System
14. Richard M. Fujimoto, Randall Guensler, Michael Hunter, Hoe Kyoung Kim, Jaesup Lee, John Leonard II, Mahesh Palekar, Karsten Schwan, Balasubramanian Seshasayee; Dynamic Data Driven Application Simulation of Surface Transportation Systems
15. Alok Chaturvedi, Angela Mellema, Sergei Filatyev, Jay Gore; DDDAS Approach to Fire and Agent Evacuation Modeling: Case Study of Rhode Island Nightclub Fire
16. James D McCalley, Vasant G Honavar, Sarah M Ryan, William Q Meeker, Ronald A Roberts, Daji Qiao, Yuan Li; Auto- Steered Information-Decision Processes for Electric System Asset Management
17. E. H. Abed, N. S. Nmachchivaya, T. J. Overbye, M. A. Pai, P. W. Sauer and A. Sussman; Data-Driven Power System Operations
18. C. Farhat, J. G. Michopoulos, F.K. Chang, L.J. Guibas, and A.J. Lew; Towards a Dynamic Data Driven System for Structural and Material Health Monitoring
19. Asad Awan, Ahmed Sameh, and Ananth Grama; The Omni Macroprogramming Environment for Sensor Networks
20. D. Knight, T. Rossman and Y. Jaluria; Evaluation of Fluid-Thermal Systems by Dynamic Data Driven Application Systems
21. Volkan Akcelik, George Biros, Andrei Draganescu, Omar Ghattas, Judith Hill, and Bart van BloemenWaanders; Inversion of Airborne Contaminants in a Regional Model
22. S. Kim, J. Chandrasekar, A. Ridley, D. S. Bernstein; Data Assimilation Using the Global Ionosphere-Thermosphere Model
23. Sai Ravela; Amplitude-Position formulation of Data Assimilation
24. Hyung-Jin Son and Theodore B. Trafalis; Detection of Tornadoes Using an Incremental Revised Support Vector Machine with Filters
25. Leana Golubchik, David Caron, Abhimanyu Das, Amit Dhariwal, Ramesh Govindan, David Kempe, Carl Oberg, Abhishek Sharma, Beth Stauer, Gaurav Sukhatme, Bin Zhang; A Generic Multi-scale Modeling Framework for Reactive Observing Systems: an Overview
26. Craig C. Douglas, Jonathan D. Beezley, Janice Coen, Deng Li, Wei Li, Alan K. Mandel, Jan Mandel , Guan Qin, and Anthony Vodacek; Demonstrating the Validity of a Wildfire DDDAS
27. J. T. Oden, K. R. Diller, C. Bajaj, J. C. Browne, J. Hazle2, I. Babujska, J. Bass, L. Demkowicz, Y. Feng, D. Fuentes, S. Prudhomme, N. Rylander, R. J. Sta_ord, and Y. Zhang; Development of a Computational Paradigm for Laser Treatment of Cancer
28. P. D. Richardson, I.V. Pivkin, G.E. Karniadakis, D.H. Laidlaw; Blood Flow At Arterial Branches: Complexities To Resolve For The Angioplasty Suite
29. José Fortes, Renato Figueiredo, Linda Hermer-Vazquez, José Príncipe and Justin C. Sanchez; A New Architecture for Deriving Dynamic Brain-Machine Interfaces
30. D. Metaxas, G. Tsechpenakis, Z. Li, Y. Huang, A. Kanaujia; Dynamically Adaptive Tracking of Gestures and Facial Expressions
31. Catriola Kennedy and Georgios Theodoropoulos; Intelligent Management of Data Driven Simulations to Support Model building in the Social sciences
32. Paul Reynolds, David Brogan, Joseph Carnahan, Yannick Loit`ere, and Michael Spiegel; Capturing Scientists' Insight for DDDAS
33. Adel Torkaman Rahmani, Vahid Rafe, Saeed Sedighian and Amin Abbaspour; An MDA-based Modeling and Design of Service-Oriented Architecture
34. Anthony Jones and Dan Cornford; Advanced Data Driven Visualisation for Geo-spatial Data

35. Liu Bo, Zhao Jun, and Qian Jixin; Design and Analysis of Test Signals for System Identification
36. XiaoDong Gao and Zhiping Fan; The Research on the Method of Process-based Knowledge catalog & Storage and its Application in Steel Product R&D
37. Emil M. Constantinescu, Adrian Sandu, Gregory R. Carmichael, Tianfeng Chai, John H. Seinfeld, and Dacian Dgaescu; Localized Ensemble Kalman Data Assimilation for Atmospheric Chemical Transport Models

Towards Dynamic Data-Driven Management of the Ruby Gulch Waste Repository*

Manish Parashar¹, Vincent Matossian¹, Hector Klie², Sunil G. Thomas²,
Mary F. Wheeler², Tahsin Kurc³, Joel Saltz³, and Roelof Versteeg⁴

¹ TASSL, Dept. of Electrical & Computer Engineering, Rutgers,
The State University of New Jersey, New Jersey, USA

{parashar, vincentm}@caip.rutgers.edu

² CSM, ICES, The University of Texas at Austin, Texas, USA

{klie, sgthomas, mfw}@ices.utexas.edu

³ Dept. of Biomedical Informatics, The Ohio State University, Ohio, USA

{kurc, jsaltz}@bmi.osu.edu

⁴ INL, Idaho, USA

roelof.versteeg@inl.gov

Abstract. Previous work in the Instrumented Oil-Field DDDAS project has enabled a new generation of data-driven, interactive and dynamically adaptive strategies for subsurface characterization and oil reservoir management. This work has led to the implementation of advanced multi-physics, multi-scale, and multi-block numerical models and an autonomic software stack for DDDAS applications. The stack implements a Grid-based adaptive execution engine, distributed data management services for real-time data access, exploration, and coupling, and self-managing middleware services for seamless discovery and composition of components, services, and data on the Grid. This paper investigates how these solutions can be leveraged and applied to address another DDDAS application of strategic importance - the data-driven management of Ruby Gulch Waste Repository.

1 Introduction

The dynamic, data driven application systems (DDDAS) paradigm is enabling a new generation of end-to-end multidisciplinary applications that are based on seamless aggregation of and interactions between computations, resources, and data. An important class of DDDAS applications include simulations of complex physical phenomena that symbiotically and opportunistically combine

* The research presented in this paper is supported in part by the National Science Foundation Grants ACI-9984357, EIA-0103674, EIA-0120934, ANI-0335244, CNS- 0305495, CNS-0426354, IIS-0430826, ACI-9619020 (UC Subcontract 10152408), ANI-0330612, EIA-0121177, SBR-9873326, EIA-0121523, ACI-0203846, ACI-0130437, CCF-0342615, CNS- 0406386, CNS-0426241, ACI-9982087, CNS-0305495, NPACI 10181410, Lawrence Livermore National Laboratory under Grant B517095 (UC Subcontract 10184497), Ohio Board of Regents BRTTC BRTT02-0003, and DOE DE-FG03-99ER2537.

computations, experiments, observations, and real-time data to provide important insights into complex systems.

In a recent DDDAS project, we have developed several key technologies to enable a new generation of data-driven, interactive and dynamically adaptive strategies for subsurface characterization and reservoir management. This “Instrumented Oil-Field” project [1, 2, 3] aimed at completing the symbiotic feedback loop between measured data and the computational models to provide more efficient, cost-effective and environmentally safer production of oil reservoirs, which can result in enormous strategic and economic benefits. Our work in this project has led to conceptual and infrastructure solutions [1, 2, 3, 4, 5, 6, 7], which include advanced multi-physics, multi-scale and multi-block numerical models as well as an autonomic DDDAS software stack. The software stack provides a middleware for autonomic DDDAS applications and consists of a Grid-based execution engine that supports self-optimizing, dynamically adaptive applications, distributed data management services for large scale data management and processing, and self-managing middleware services for seamless discovery, access, interactions and compositions of components, service and data on the Grid.

In this work, we look at the application of these technologies in the more general class of knowledge-based, data-driven subsurface management applications. This paper is concerned with the problem of dynamic data-driven waste management at the Ruby Gulch Waste Repository, and more specifically, the Gilt-Edge site. It investigates how the DDDAS models, methodologies and software stack can be applied to address this challenging application.

2 The Dynamic Data-Driven Waste Management Problem: The Ruby Gulch Waste Repository

The Gilt Edge Mine is located near Deadwood, South Dakota. Mining for gold and silver at this site occurred from 1880-1999. In 1999 the Dakota Mining Corporation (which operated the mine through its subsidiary, Brohm Mining Company) declared bankruptcy, and the site reverted to the state of South Dakota. Mining activities had resulted in several negative environmental impacts on the site. One of the main environmental issues was the presence of multiple sources of ARD (Acid Rock Drainage). The primary source was the Ruby Waste Rock Repository. This repository is a valley in which mine rock was disposed of post leaching. It contains approximate 11 million cubic yard of waste rock. As ARD

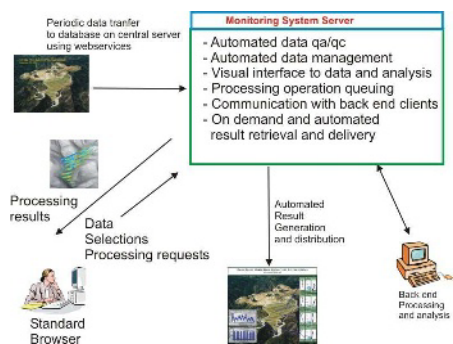


Fig. 1. Gilt Edge Monitoring System Server

flowing from this repository would severely impact drinking water quality downstream of the site, this ARD needs to be captured and treated.

In order to minimize the amount of water coming out of this repository a ROD (Record of Decision) for this repository called for the emplacement of a cap over the site. As EPA had interest in the monitoring of the performance of this cap, a monitoring system was designed and installed by scientists from the Idaho National Laboratory (INL) [8, 9]. The monitoring system autonomously collects continuous data using the following sensors: (1) a weather station operated by SDENR (South Dakota Department of Environment and Natural Resources); (2) an outflow meter at the bottom of the Ruby Repository; (3) temperature sensors in four well boreholes; (4) advanced tensiometers are located within boreholes and measure matrix potential (related to water saturation); (5) a multi-electrode resistivity system.

In addition to the autonomously collected data, gas-ports and porous-ceramic cup-lysimeters in the wells are sampled monthly (starting in June 2004). Gas analysis is done in the field for CO₂ and O₂. Water samples are sent to a laboratory for analysis. In addition to this data there is a substantial amount of historical data (collected by BMC and DENR), which has been aggregated into the database. Figure 1 illustrates the monitoring process at the Gilt Edge mine.

3 Models, Methods and Middleware for Data-Driven Subsurface Management Applications

3.1 The Integrated Parallel Accurate Reservoir Simulator (IPARS)

IPARS represents a new approach to parallel reservoir simulator development, emphasizing modularity, code portability to many platforms, ease of integration and inter-operability with other software. It provides a set of computational features such as memory management for general geometric grids, portable parallel communication, state-of-the-art non-linear and linear solvers, keyword input, and output for visualization. A key feature of IPARS is that it allows the definition of different numerical, physical, and scale models for different blocks in the domain (i.e., multi-numeric, multi-physics, and multi-scale capabilities). A more technical description of IPARS and its applications can be found in [10].

3.2 Optimization Algorithms

Novel stochastic optimization algorithms [4, 5] have been included in the framework, namely, the Very Fast Simulated Annealing (VFSA), the Finite Difference Stochastic Approximation (FDSA) and the Simultaneous Perturbation Stochastic Approximation (SPSA). The VFSA and SPSA, in particular, have potentials for large scale implementations. The critical issue here, is that parameter estimation may involve several hundreds of thousands of variables and the objective function evaluation is a highly demanding process since it involves a full simulation run. This also rules out the possibility for using gradient-based methods,

especially in a multi-model environment that systematically aims at different physics and algorithms for which sensitivity coefficients are not trivial either to compute or reformulate.

3.3 Storage and Management of Large Volumes of Data

Effective solutions to the waste management problem targeted in this work will require gleaning and extracting information from results of optimization processes using complex numerical models and from data gathered by field sensors. Datasets generated by an optimization run consists of the values of the input and output parameters along with the output from simulations of a numerical model of the physical domain. Datasets gathered from field sensors consist of readings obtained from each sensor, the location of the sensor, and the date of the reading. This information provides a dynamically updated (as more readings are obtained) historical record of the field under study. By maintaining simulation and sensor datasets, a large-scale dynamic knowledge base can be created. This knowledge base can be used to speed up the execution of optimization runs, to carry out post-optimization analyses, to refine numerical models using field data, and to control where and how much field data should be collected, thus implementing a dynamic, data-driven application system approach. Common types of queries against these datasets include computing data subsets via range queries, aggregations such as counts, averages on over regions of meshes, and differences between regions of interest on multi-resolution datasets. With the help of inexpensive disk-based storage, we are seeing the emergence of large scale, hierarchical storage platforms with varying capacity/bandwidth (from larger, slower disk pools to smaller, faster disks to memory on cluster) and distance from compute nodes. A challenging issue is to be able to use different levels of storage and computing hierarchy in a coordinated way to maximize the bandwidth of data retrieval and processing. A number of strategies, such as multi-level hierarchical indexing [11], partial replication [12], caching [13], and adaptive data redistribution can be employed.

To support the knowledge base, we make use of three middleware systems to support the data management and processing requirements as described above of optimization based studies for waste management application. STORM [14] is a service-oriented middleware that supports data select and data transfer operations on scientific datasets, stored in distributed, flat files, through an object-relational database model. Mobius [15] provides support for management of metadata definitions, on-demand database creation, and federation of distributed XML databases. It uses XML schemas to define the structure of data elements and XML documents to represent instances of data elements. DataCutter [16] provides a coarse-grained data flow system and allows combined use of task- and data-parallelism. In DataCutter, application processing structure is implemented as a set of components, called *filters*, that exchange data through a *stream* abstraction.

3.4 Autonomic Grid Middleware Substrate

Emerging knowledge-based and dynamic data-driven subsurface management and control applications, such as the applications described in this paper, combine computations, experiments, observations, and real-time data, and are highly heterogeneous and dynamic in their scales, behaviors, couplings and interactions. AutoMate [17], an Autonomic Computational Engine for management and control, investigates conceptual models and implementation architectures to address these challenges and enable the development and execution of such self-managing Grid applications. Key components include:

- The *Seine/MACE Computational Engine* [18] that implements a dynamic geometry-based shared space interaction model to support the dynamic and complex communication and coordination patterns required by the multi-block parallel multi-block simulations.
- The *Accord Programming System* [19] that enables the definition of autonomic components and the dynamic composition, management and optimization of these components using externally defined rules and constraints. Autonomic components in *Accord* export sensors and actuators for external monitoring, control and adaptation.
- The *Autonomic Runtime Environment* [20] provides policies and mechanisms for both “system sensitive” and “application sensitive” runtime adaptations to manage the heterogeneity and dynamism of the applications as well as Grid environments.
- The *Content-based Grid Middleware Substrate* [17] that supports autonomic application behaviors and interactions, and to enable simulation components, sensors/actuators, data archives and Grid resources and services to seamlessly interact as peers. Key components of the middleware include the Meteor, a decentralized infrastructure for decoupled associative interactions, the Squid content-based routing engine and decentralized information discovery service, and the Pawn peer-to-peer messaging substrate.
- The *Discover Collaboratory* [21] that provides collaborative problem solving environment and enables geographically distributed scientists and engineers to collaboratively monitor, interact with, and control high performance applications in a truly pervasive manner using portals.

4 Dynamic Data-Driven Waste Management: Modeling the Gilt-Edge Site with Dynamic Data

In order to make better predictions from the measurements at the Gilt Edge site, the first task is to develop a model that suitably explains the observations from the experiments at the waste repository. For instance, it was observed that there exists a diurnal/seasonal variation in the outflow measurements. Using IPARS, a system of modified air-water equations can be used to model the problem. This solution takes into account that water can exist in the air phase as vapor,

and can explain the diurnal variations qualitatively. An example water pressure profile reproduced by the model is shown in Figure 2.

Once the predicted model is calibrated, the next challenge is to determine the physical parameters of the site such as permeability, porosity and capillary pressure in order to reproduce the exact measured outflows at the site. This task can be modeled as a parameter estimation problem using the numerical model of the environment. This is where an efficient optimization method such as SPSA or VFSA plays a significant role. At present, the SPSA method is being implemented on the hydrology model in IPARS. Using the autonomic computational engine, the execution of IPARS and the optimization methods can be dynamically orchestrated in a Grid environment [1, 2].

To perform parameter estimation a mismatch function based on the difference of measured and calculated outflow of water at a specified location at the site is posed. Similarly, an objective function based on maximum cleanup rate can be designed for optimal site management. To extend these optimizations for data assimilation, the numerical models implemented using IPARS would need to be refined using dynamic data from outflow measurements at the site. To achieve this, an optimal control scheme should be formulated to accommodate dynamic changes into the parameters (i.e., properties) and state variables (e.g., saturations, pressures, temperatures) of the model. Increasing understanding of the physical model and the need to respond more quickly to observations leads to metamodels (surrogate models) or reduced models. These simpler models mimic the behavior of the original predictive model given by IPARS.

Accurate model prediction and optimization capabilities in conjunction with the Grid middleware and data management tools described in Section 3 makeup the fundamental components of the The Dynamic Data-Driven Waste Management framework proposed in this work (see Figure 3). The framework adds autonomic decision making and control capabilities to the monitoring process.

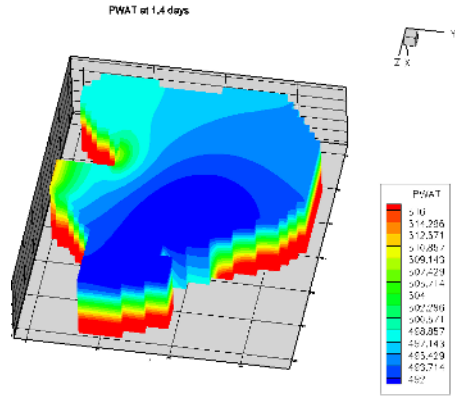


Fig. 2. Example of an air-water simulation to predict water pressure profile

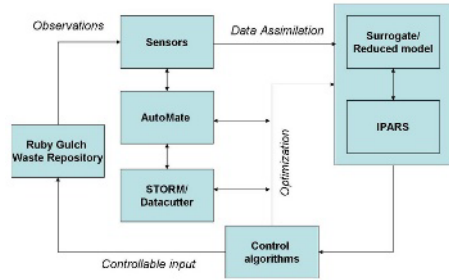


Fig. 3. The Dynamic Data-Driven Waste Management framework

The autonomic computational engine and middleware service provide the infrastructure for: (1) enabling the efficient, large scale, dynamically adaptive multi-block IPARS simulations; (2) for discovering, aggregating and assimilating data from the sensors at the remote Gilt-Edge site and dynamically injecting it into the simulation processes as required; (3) selecting and invoking appropriate optimization services; (4) enabling dynamic composition of services to realize data driven workflows on the Grid; and (4) enabling remote collaborative and interactive access to the simulations and the data using pervasive portals.

The estimation of physical properties of the environment involves search of a parameter space and requires data from outflow measurements dynamically drive the simulation of the numerical models and parameter space search. A knowledge base can be created from the datasets that are generated (via simulations or field measurements) and referenced in this application to speed up the optimization process. Mobius can be used to manage metadata associated with distributed datasets in the knowledge base. At any given step during optimization the knowledge base can be queried to see if a given step, or a subset of numerical simulations at that step, has been already evaluated. STORM and DataCutter can be employed to support queries into distributed collections of large datasets stored as a collection of files. For instance, in post-optimization analyses, a user may want to compare and correlate a subset of results obtained from one optimization run with results from another set of optimization runs.

5 Conclusion

Our previous work in the Instrumented Oil-Field DDDAS project has developed advanced numerical models and a suite of software components, which provide support for 1) execution of dynamically adaptive applications in a Grid environment, 2) management and manipulation of large distributed datasets, and 3) seamless discovery of, interaction with, and composition of application components, services, and data in the Grid. We believe that these advanced numerical models and tools can be applied to other types of data-driven subsurface management applications. In this paper we investigated how these technologies can be leveraged to enable data-driven management of Ruby Gulch Waste Repository.

References

1. Parashar, M., Klie, H., Catalyurek, U., Kurc, T., Matossian, V., Saltz, J., Wheeler, M.: Application of grid-enabled technologies for solving optimization problems in data-driven reservoir studies. In: Proceedings of the Workshop on Distributed Data Driven Applications and Systems, International Conference on Computational Science 2004 (ICCS 2004). Volume 3038., Krakow, Poland (2004) 805 – 812
2. Parashar, M., Matossian, V., Bangerth, W., Klie, H., Rutt, B., Kurc, T., Catalyurek, U., Saltz, J., Wheeler, M.: Towards dynamic data-driven optimization of oil well placement. In: Proceedings of the Workshop on Distributed Data Driven Applications and Systems, International Conference on Computational Science 2005 (ICCS 2005). Volume 3514-3516., Atlanta, USA (2005) 656 – 663

3. Klie, H., Bangerth, W., Gai, X., Wheeler, M.F., Stoffa, P., Sen, M., Parashar, M., Catalyurek, U., Saltz, J., Kurc, T.: Models, methods and middleware for grid-enabled multiphysics oil reservoir management. *Engineering with Computers*, Springer-Verlag (2006)
4. Matossian, V., Bhat, V., Parashar, M., Peszynska, M., Sen, M., Stoffa, P., Wheeler, M.F.: Autonomic oil reservoir optimization on the grid. *Concurrency and Computation: Practice and Experience* **17** (2005) 1–26
5. Bangerth, W., Klie, H., Matossian, V., Parashar, M., Wheeler, M.F.: An autonomic reservoir framework for the stochastic optimization of well placement. *Cluster Computing: The Journal of Networks, Software Tools, and Applications* **8** (2005) 255–269
6. Kurc, T., Catalyurek, U., Zhang, X., Saltz, J., Martino, R., Wheeler, M., Peszyńska, M., Sussman, A., Hansen, C., Sen, M., Seifoullaev, R., Stoffa, P., Torres-Verdin, C., Parashar, M.: A simulation and data analysis system for large scale, data-driven oil reservoir simulation studies. *Concurrency and Computation: Practice and Experience*. **17** (2005) 1441–1467
7. Parashar, M., Muralidhar, R., Lee, W., Wheeler, M., Arnold, D., Dongarra, J.: Enabling interactive and collaborative oil reservoir simulations on the grid. *Concurrency and Computation: Practice and Experience* **17** (2005) 1387–1414
8. Versteeg, R., Wangerud, K., et al.: Managing a capped acid rock drainage (ard) repository using semi-autonomous monitoring and modeling. In: ICARD 2006, St. Louis, Missouri (2006)
9. Wangerud, K., Versteeg, R., et al.: Insights into hydrodynamic and geochemical processes in a valley-fill ard waste-rock repository from an autonomous multi-sensor monitoring system. In: ICARD 2006, St. Louis, Missouri (2006)
10. (Ipars: Integrated parallel reservoir simulator) The University of Texas at Austin, <http://www.ices.utexas.edu/CSM>.
11. Zhang, X., Pan, T., Catalyurek, U., Kurc, T., Saltz, J.: Serving queries to multi-resolution datasets on disk-based storage clusters. In: Proceedings of 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CC-Grid2004), Chicago, IL (2004)
12. Weng, L., Catalyurek, U., Kurc, T., Agrawal, G., Saltz, J.: Servicing range queries on multidimensional datasets with partial replicas. In: Proceedings of the 5th IEEE/ACM International Symposium on Cluster Computing and the Grid (CC-Grid 2005). (2005)
13. Deshpande, P.M., Ramasama, K., Shukla, A., Naughton, J.F.: Caching multidimensional queries using chunks. In: ACM SIGMOD Record, Vol. 27, No. 2. (1998) 259–270
14. Narayanan, S., Kurc, T., Catalyurek, U., Zhang, X., Saltz, J.: Applying database support for large scale data driven science in distributed environments. In: Proceedings of the Fourth International Workshop on Grid Computing (Grid 2003), Phoenix, Arizona (2003) 141–148
15. Hastings, S., Langella, S., Oster, S., Saltz, J.: Distributed data management and integration: The mobius project. In: GGF Semantic Grid Workshop 2004, GGF (2004) 20–38
16. Beynon, M.D., Kurc, T., Catalyurek, U., Chang, C., Sussman, A., Saltz, J.: Distributed processing of very large datasets with DataCutter. *Parallel Computing* **27** (2001) 1457–1478

17. Parashar, M., Liu, H., Li, Z., Matossian, V., Schmidt, C., Zhang, G., Hariri, S.: Automate: Enabling autonomic grid applications. *Cluster Computing: The Journal of Networks, Software Tools, and Applications, Special Issue on Autonomic Computing* **9** (2006)
18. Zhang, L., Parashar, M.: Seine: A dynamic geometry-based shared space interaction framework for parallel scientific applications. *Concurrency and Computations: Practice and Experience* (2006)
19. Liu, H., Parashar, M.: Accord: A programming framework for autonomic applications. *IEEE Transactions on Systems, Man and Cybernetics, Special Issue on Engineering Autonomic Systems* (2006)
20. Chandra, S., Parashar, M., Yang, J., Zhang, Y., Hariri, S.: Investigating autonomic runtime management strategies for samr applications. *International Journal of Parallel Programming* **33** (2005) 247–259
21. Mann, V., Parashar, M.: DISCOVER: A computational collaboratory for interactive grid applications. In Berman, F., Fox, G., Hey, T., eds.: *Grid Computing: Making the Global Infrastructure a Reality*, John Wiley and Sons (2003) 727–744

Dynamic Contaminant Identification in Water

Craig C. Douglas^{1,2}, J. Clay Harris³, Mohamed Iskandarani⁴,
Chris R. Johnson⁵, Robert J. Lodder³, Steven G. Parker⁵, Martin J. Cole⁵,
Richard Ewing⁶, Yalchin Efendiev⁶, Raytcho Lazarov⁶, and Guan Qin⁶

¹ University of Kentucky, Department of Computer Science, 773 Anderson Hall,
Lexington, KY 40506-0046, USA

² Yale University, Department of Computer Science, P.O. Box 208285
New Haven, CT 06520-8285, USA
douglas-craig@cs.yale.edu

³ University of Kentucky, Department of Chemistry, Lexington, KY, 40506 USA
claymay27@gmail.com, lodder@contactincontext.org

⁴ University of Miami, Rosenstiel School of Marine and Atmospheric Science,
4600 Rickenbacker Causeway, Miami, FL 33149-1098, USA
mohamed.iskandarani@rsmas.miami.edu

⁵ University of Utah, Scientific Computing and Imaging Institute,
Salt Lake City, UT 84112, USA
{crj, sparker, mjc}@cs.utah.edu

⁶ Texas A&M University, Institute for Scientific Computation, 612 Blocker,
3404 TAMU, College Station, TX 77843-3404, USA
{richard_ewing, guan.qin}@tamu.edu, {efendiev, lazarov}@math.tamu.edu

Abstract. We describe how we plan to convert a traditional data collection sensor and ocean model into a DDDAS enabled system for identifying contaminants and then reacting with different models, simulations, and sensing strategies in a symbiotic manner. The sensor is just as useful in water as it would be on Mars for material identification. A successful terrestrial application of the sensor will lead to many new applications of the device and possible technology transfer to the private sector.

1 Introduction

The Solid-State Spectral Imager (SSSI) is a new instrument to gather hydrological and geological data and to perform chemical analyses. It is suitably small and light to mount in remote sensing applications, and can scan ranges of up to 10 meters. Using a laser-diode array, photodetectors, and on-board processing, the SSSI combines innovative spectroscopic integrated sensing and processing with a hyperspace data analysis algorithm [1].

Ultraviolet (UV), visible, and near-infrared laser diodes illuminate target points using a precomputed sequence, and a photodetector records the amount of reflected light. For each point illuminated, the resulting reflectance data is processed to separate the contribution of each wavelength of light and classify the substances present.

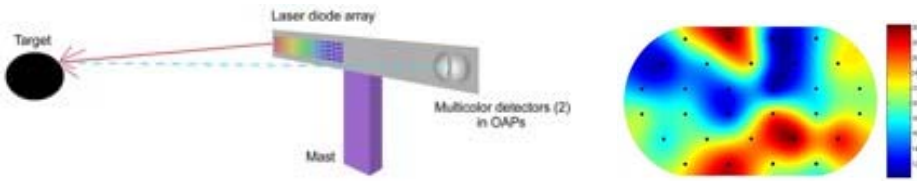


Fig. 1. SSSI emitter and collected spectrum

Several prototype implementations of SSSI have been developed and are being tested at the University of Kentucky. A full-scale implementation of SSSI is being designed with 25 lasers in discrete wavelengths between 300 nm and 2400 nm with 5 rows of each wavelength. This full-scale version is designed to consume less than 4 Watts and weigh less than 600 grams. The rugged laser diodes and detectors allow SSSI to be packaged in a small, rugged, space qualified package. For water monitoring in the open ocean, imaging capability is not needed, and a single row of diodes (with one diode at each frequency) is sufficient, and power consumption of the optical system can be reduced to approximately one watt.

The SSSI combines near-infrared, visible, and ultraviolet spectroscopy with a powerful statistical classification algorithm to detect and identify contaminants in water. Virtually every organic compound (e.g., polycyclic aromatic hydrocarbons, paraffins, carboxylic acids, and sulfonic acids) has a near-IR spectrum that can be measured, including two classes of terrestrial biomarkers, lipids, and amino acids. Near-infrared spectra consist of overtones and combinations of fundamental mid-infrared bands, giving near-infrared spectra a powerful ability to identify organic compounds while still permitting some penetration of light into samples [2].

To further increase the signal-to-noise ratio, the SSSI uses Walsh-Hadamard or CRISP encoding sequences of light pulses. In a Walsh-Hadamard sequence multiple laser diodes illuminate the target at the same time, increasing the number of photons received at the photo detector. The Walsh-Hadamard sequence can be demultiplexed to individual wavelength responses with a matrix-vector multiply [3]. Benefits of generating encoding sequences by this method include equivalent numbers of on and off states for each sequence and a constant number of diodes in the on state at each resolution point of a data acquisition period.

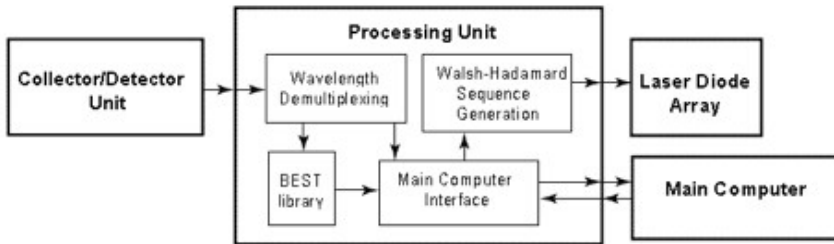


Fig. 2. SSSI processing

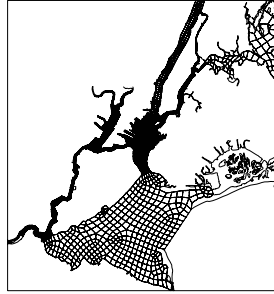


Fig. 3. Spectral element grid showing elemental partition of the New York/Newark Bay estuarine system

CRISP encoding uses orthogonal pseudorandom codes with unequal numbers of on and off states. The duty cycle of each code is different, and the codes are selected to deliver the highest duty cycles at the wavelengths where the most light is needed and lowest duty cycle where the least light is needed to make the sum of all of the transmitted (or reflected) light from the samples proportional to the analyte concentration of interest.

The initial deployment of the sensor and model will focus on estuarine regions where water quality monitoring is critical for human health and environmental monitoring. The authors will capitalize on an existing configuration of the model to the Hudson-Raritan Estuary to illustrate the model’s capabilities. As shown in Fig. 3 the model domain includes the Lower and Middle Hudson River, the Hudson-Raritan Estuary, Newark Bay and Long Island sound. In the initial experimentation stage only a portion of the grid will be for fast prototyping of the different elements in the DDDAS system.

The forward model is based on the two-dimensional Spectral Element Ocean Model (SEOM-2D) which solves the shallow water equations:

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{f} \times \mathbf{u} + g \nabla \eta = \frac{\tau_w - \tau_d}{\rho h} + \frac{\nabla \cdot (h\nu \nabla \mathbf{u})}{h} \tag{1}$$

$$h_t + \nabla \cdot (h\mathbf{u}) = Q \tag{2}$$

where $h = H + \eta$ is the layer thickness, and H and η are the resting depth and interface displacement, respectively, \mathbf{u} is the depth-average velocity; g is the gravity coefficient, \mathbf{f} is the Coriolis parameter, ν the viscosity, τ_w the surface wind stress, τ_b the bottom drag, and Q is an area mass source. An advection-diffusion equation tracks the evolution of passive tracers:

$$T_t + \mathbf{u} \cdot \nabla T = \frac{\nabla \cdot (\alpha h \nabla T)}{h} \tag{3}$$

where α is the diffusion coefficient, and T stands for a generic tracer. The model can be forced through winds, tides, and lateral injection of mass at inflow boundaries (e.g. river input).

The spectral element discretization is an h - p type finite element method which relies on relatively high degree (5-8th) polynomials to approximate the solution within each element. The main features of the spectral element method are: geometric flexibility due to its unstructured grids, its dual paths to convergence: exponential by increasing polynomial degree or algebraic via increasing the number of elements, dense computational kernels with sparse inter-element synchronization, and excellent scalability on parallel machines.

A sample tidal calculation is performed using a grid that encompasses the Newark/New York bays regions, the Long-Island sound, and a substantial portion of the Hudson River. The model is forced with tidal elevation obtained from tide gauges located on the eastern edge of the Long-Island sound and in Sandy Hook.

The SSSI is reprogrammable in the field. When an interesting chemical trace is discovered, the reaction from the application overseeing the SSSI is two-fold: (a) invoke an appropriate application, and (b) request that the SSSI look for specific other chemical traces. There is a symbiotic relationship between the sensor network and the application simulation that is typical in a DDDAS.

Consider finding gasoline or diesel fuel in a body of water. This can be a sign of innocuous pollution from a boat. Depending on what other traces are found, it could be an indication that a boat sank recently nearby. The SSSI needs to be reprogrammed in the latter case and a search and locate application must be invoked to find the sunken boat. Emergency services, the coast guard, and the news media may also need to be automatically informed of progress.

The SSSI has a modest amount of memory and computing capacity on board. Some of the computing and decision making will be put onto the SSSI over time, thus reducing the amount of time needed to reprogram the device.

2 Data Assimilation and Accurate Predictions

Data collection is initiated by a signal sent to the serial interface of the SSSI. All data is collected by the SSSI using a phototransistor connected to an operational amplifier circuit. The analog signal is converted with an on-board 0.5-5 V analog to digital converter at 12 bits. Each scan consists of 256 data points collected in both the on and off states of 25 Hadamard encoded light sequences. The result is 50 total states with 12800 data points collected for each scan. The corresponding values of the on and off states for each Hadamard coded light sequence are subtracted to remove ambient light from the data. After subtraction, the resulting 256 data points from each of the 25 Hadamard coded light sequences are then averaged to obtain 25 16 bit intensity values. The final 25 16 bit resulting values are exported to Matlab via the serial connection to a graphical user interface where data undergoes a reverse Hadamard transform to obtain intensity values for each of the 25 diodes.

A single scan with MatLab processing takes less than 300 ms. The switching speed of our transistors within the SSSI is significantly slow that this prototype requires a 5 μ s delay before each datum reading for signal stabilization after the

lights have switched states. Both times will be significantly speeded up if we move to a commercial quality device.

We can use the data to improve our prediction of the contaminant transport by updating the initial conditions. Here, initial condition refers to the concentration distribution at some previous time step. This update reduces the computational errors associated with incorrect initial data and improves the predictions. We consider contaminant transport described by (1)-(3). Initial data is sought in a finite dimensional space. Using the first set of measurements, the approximation of the initial data is recovered. As new data are incorporated into the simulator, the initial data is updated using an objective function. We note that the formulated problem is ill-posed because there are fewer sensors than the finite dimensional space describing the initial data. Consequently, the objective function is set up based on both a measurement error as well as a penalization term that depends on the prior knowledge about the solution at previous time steps (or initial data). The prior information is refreshed using the updated initial data. The penalization constants depend on time of update and can be associated with the relative difference between simulated and measured values. In the simulations, both the prior and penalization constants change in time.

To account for the errors (uncertainties) associated with sensor measurements, we consider an initial data update within a Bayesian framework. The posterior distribution is set up based on measurement errors and prior information. This posterior distribution is complicated and involves the solutions of partial differential equations. We could use a Metropolis-Hasting Markov chain Monte Carlo (MCMC) method to generate samples from the posterior distributions. However, a sampling with MCMC is expensive since it requires iterative steps and the acceptance rate is typically low. We developed an approach that combines least squares with a Bayesian approach that gives a high acceptance rate. In particular, we can prove that rigorous sampling can be achieved by sampling the sensor data from the known distribution, thus obtaining various realizations of the initial data. Our approach has similarities with the Ensemble Kalman Filter approach, which can also be adapted to an initial data update. These issues will be discussed in detail elsewhere.

3 Chemical Identification Process

A programmable, networked, portable low-cost mil-spec sensor and network for DDDAS in extreme aqueous environments must be able to perform chemical analyses to be effective in terrorist attack and accident scenarios. Most oil sensing in the oceans is done by remote sensor systems [4].

A network of sensors immersed in the ocean water (either on fixed buoys or as roving sensors) eliminates many of the problems with remote sensing. Bad weather does not affect immersed sensors. A roving sensor can be programmed to investigate beaches, weeds or debris. The SSSI is laser fluorosensor when a filter is placed over the detector, so it can positively discriminate oil on most

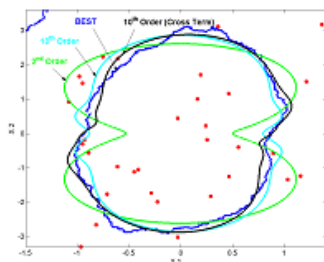


Fig. 4. Identification

backgrounds. Light scattering measurements reveal droplet size, and spectral transmission and reflectance reveal droplet chemistry.

Once the spectrum of a sample has been collected, it must be classified to determine the substance present. The Bootstrap Error-adjusted Single-sample Technique (BEST) [5] is the analytical basis of SSSI, and the foundation for the chemical library. Spectra recorded at n wavelengths are represented as single points in a n -dimensional hyperspace. In this scheme, similar samples produce similar spectra that project as “probability orbitals” or “clusters” into similar regions of hyperspace. The BEST metric is a clustering technique for exploring these distributions of spectra in hyperspace.

A sample spectrum is compared to each substance in a biogeochemical and industrial library based on its direction and distance, measured in standard deviation (SD) units, from the known substances. BEST handles asymmetric standard deviations surrounding each substance nonparametrically, allowing very precise discrimination. A sample within 3 SD units of a substance is considered to be composed of the matching substance. Any substance more than 3 SD units away from any known substance is considered an unknown substance.

For a given library entry, the BEST algorithm can be suitably approximated using multiple linear regression (MLR) to substantially reduce computational requirements (see Fig. 4). In this implementation, BEST SD units are precalculated before the SSSI is deployed in a large number of directions from the population means, and MLR is used to fit the standard deviation contours as a function of direction. With a sufficient number of terms (in the example, 10th order with cross terms), the MLR version of the algorithm can predict BEST distances to within 5% of the true value.

Oil droplets can travel nearly anywhere in the ocean. The droplet size exerts a major effect on droplet motion [6]. The rise velocity of oil droplets extends from about 2.5×10^{-7} m/s for a diameter of $2 \mu\text{m}$ to 4.3×10^{-3} m/s for a diameter of $260 \mu\text{m}$. Droplets traveling at 2.5×10^{-7} m/s will ascend only 0.001 m and 0.02 m, over periods of 1 hour and 24 hours, while over equivalent periods, droplets ascending at 4.3×10^{-3} m/s will climb 15 m and 370 m. In the meantime, a vertical diffusivity of $51 \text{ cm}^2/\text{s}$ will distribute oil droplets (equally upward and downward) about 6 m and 30 m over the same time. Therefore, the

smallest oil droplets act as though they are neutrally buoyant (transported only by diffusion), while the largest droplets are advected largely by their buoyancy.

Using multiple linear regression the BEST classification algorithm can be performed in situ, allowing a rover to classify many samples, only notifying the simulation when an interesting substance is found. An initial library can be computed based on substances likely to be found in the target environment. When a substance unknown to the BEST library is found, the sensor can sample nearby points with similar spectra to create a new library entry for the new substance. Scientists can determine the type of substance present by further analyzing raw spectra of the substance provided by SSSI and by using data from their other instruments, apply these data to update the simulation. The SSSI chemical library will comprise substances expected to be in the environment in which the SSSI operates.

4 Matlab and SCIRun Environments

The SCIRun-Matlab interface is designed such that SCIRun [7] detects at runtime whether Matlab is available. Hence, SCIRun does not have to be linked against any of the Matlab libraries. The way the interfacing is accomplished is through a virtual shell. SCIRun accesses Matlab through stdin and stdout and files that are written to a temporary directory. The whole process of translating, saving, and opening files is hidden from the user and is initiated automatically. Data translation is also seamless for things like 0 or 1 based indices.

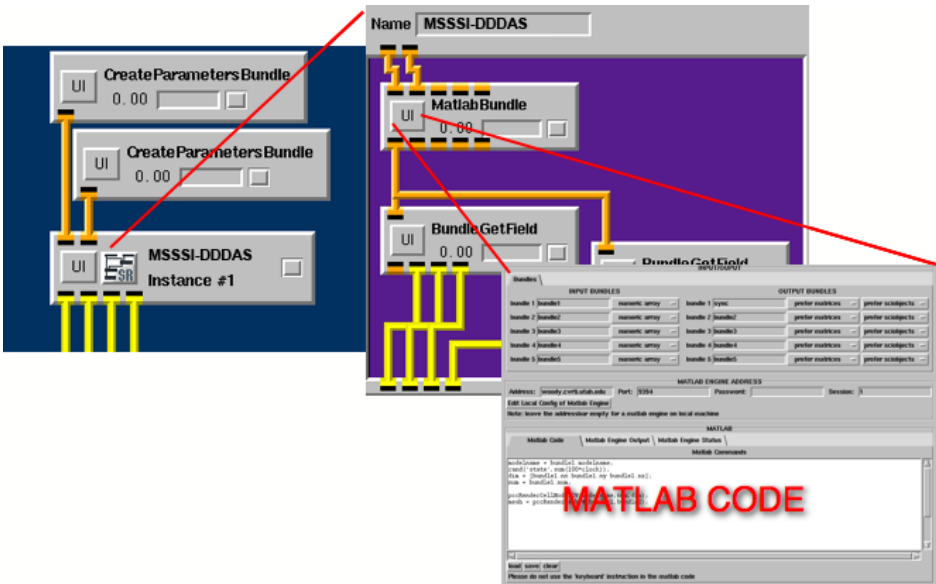


Fig. 5. SCIRun running the SSSI-cid module

The module is designed in such a way that once the Module is executed it will keep Matlab running in its internal engine, hence re-executing the module will allow to access the variables that were left by a previous execution cycle. Hence the Matlab Engine can be used as well for iterative processes. An example of the matlabinterface module is depicted in Fig. 5. The figure shows how matlab is integrated into the dataflow structure of SCIRun. Once the "Matlab" module receives all the dataflow objects that are connected to it, the specific code in Matlab is executed and now dataflow objects are created for the dataflow downstream.

Hence, moving the current Matlab interface for the SSSI sensor to a problem solving environment like SCIRun is trivial.

5 Conclusions

We described how we plan to convert a traditional data collection sensor and ocean model into a DDDAS enabled system for identifying contaminants and then reacting with different models, simulations, and sensing strategies in a symbiotic manner. A drone is being built so that the SSSI will be mobile. We are already able to make measurements and are proceeding to program the system for remote sensing and steering. Libraries will be created for interesting contaminants during the coming year that we will use to reprogram the SSSI dynamically while we switch to an appropriate simulation for the contaminants identified to explore what else might be in the vicinity of the SSSI.

References

1. Lowell, A., Ho, K.S., Lodder, R.A.: Hyperspectral imaging of endolithic biofilms using a robotic probe. *Contact in Context* **1** (2002) 1–10
2. Dempsey, R.J., Davis, D.G., R. G. Buice, J., Lodder, R.A.: Biological and medical applications of near-infrared spectrometry. *Appl. Spectrosc.* **50** (1996) 18A–34A
3. Silva, H.E.B.D., Pasquini, C.: Dual-beam near-infrared Hadamard. *Spectrophotometer Appl. Spectrosc.* **55** (2001) 715–721
4. Fingas, M.F., Brown, C.E.: Review of oil spill remote sensing. In: *Spillcon 2000*, Darwin, Australia (2000)
5. Dieter, W., Lodder, R.A., James E. Lumpp, J.: Scanning for extinct astrobiological residues and current habitats (SEARCH) using integrated computational imaging. *IEEE Aerospace and Electronic Systems* (2006) (in press)
6. (OSB), O.S.B.: *Oil Spill Dispersants: Efficacy and Effects* (2005). The National Academies Press, Washington, DC (2005)
7. Johnson, C.R., Parker, S., Weinstein, D., Heffernan, S.: Component-based problem solving environments for large-scale scientific computing. *Concur. Comput.: Practice and Experience* **14** (2002) 1337–1349

An Adaptive Cyberinfrastructure for Threat Management in Urban Water Distribution Systems

Kumar Mahinthakumar¹, Gregor von Laszewski², Ranji Ranjithan¹, Downey Brill¹, Jim Uber³, Ken Harrison⁴, Sarat Sreepathi¹, and Emily Zechman¹

¹ North Carolina State University, Raleigh, NC, USA
{gmkumar, ranji, brill, sarat_s, emzechma}@ncsu.edu

² University of Chicago, Chicago, IL, USA
gregor@mcs.anl.gov

³ University of Cincinnati, Cincinnati, OH, USA
jim.uber@uc.edu

⁴ University of South Carolina, Columbia, SC, USA
harriskw@engr.sc.edu

Abstract. Threat management in drinking water distribution systems involves real-time characterization of any contaminant source and plume, design of control strategies, and design of incremental data sampling schedules. This requires dynamic integration of time-varying measurements along with analytical modules that include simulation models, adaptive sampling procedures, and optimization methods. These modules are compute-intensive, requiring multi-level parallel processing via computer clusters. Since real-time responses are critical, the computational needs must also be adaptively matched with available resources. This requires a software system to facilitate this integration via a high-performance computing architecture such that the measurement system, the analytical modules and the computing resources can mutually adapt and steer each other. This paper describes the development of such an adaptive cyberinfrastructure system facilitated by a dynamic workflow design.

1 Introduction

Urban water distribution systems (WDSs) are vulnerable to accidental and intentional contamination incidents that could result in adverse human health and safety impacts [1]. The pipe network in a typical municipal WDS includes redundant flow paths to ensure service when parts of the network are unavailable, and is designed with significant storage to deliver water during daily peak demand periods. Thus, a typical network is highly interconnected and experiences significant and frequent fluctuations in flows and transport paths. These design features unintentionally enable contamination at a single point in the system to spread rapidly via different pathways through the network, unbeknown to consumers and operators due to uncertainty in the state of the system. This uncertainty is largely a function of spatially and temporally varying water usage. When a contamination event is detected via the first line of defense, e.g., data from a water quality surveillance sensor network and reports from consumers,

the municipal authorities are faced with several critical questions as the contamination event unfolds: Where is the source of contamination? When and for how long did this contamination occur? Where should additional hydraulic or water quality measurements be taken to pinpoint the source more accurately? What is the current and near future extent of contamination? What response action should be taken to minimize the impact of the contamination event? What would be the impact on consumers by these actions?

Real-time answers to such complex questions will present significant computational challenges. We envision a dynamic integration of computational approaches (the conjunctive use of simulation models with Bayesian and optimization methods) and spatial-temporal data management methods, enabled within a grid-based software framework. Each component in the cyberinfrastructure can be represented at multiple scales within a hierarchy, and must mutually adapt according to the levels within the hierarchy of the associated components. For example, the level of accuracy of the chosen model must appropriately match the degree of refinement in the available data or steer the measurement system to obtain additional data. Similarly, the degree of sophistication of the search method must appropriately match the available computing resources on the grid or steer the grid brokering system to obtain more resources. Thus the cyberinfrastructure must facilitate dynamic interactions among its components through a mutually adaptive and steerable software system. This cyberinfrastructure will be enabled by recently developed grid-based resources and high performance networks. Developing and demonstrating such a cyberinfrastructure for adaptive contaminant assessment in municipal WDSs is the overall goal of this research.

2 Technical Objectives

The primary technical objectives of this DDDAS [2] research are to:

- Develop simulation procedures and search algorithms that can adapt to changing conditions including data and computational resources. These procedures are part of a strategic decision-making framework for source characterization, hydraulic control, and activation and deployment of sensors for threat management in urban water distribution systems.
- Implement a grid-enabled dynamic work flow engine that can adaptively assemble and drive various data, computational, and computer resource components for changing conditions and demands. This engine will enable tractable execution of the algorithms as well as real-time processing of sensed data to help make rapid, tactical decisions related to hydraulic control and confirmatory sampling.
- Test and evaluate the work flow engine and the associated components for hydraulic control, water quality and hydraulic sensor network design and confirmatory sampling procedures for an array of threat management scenarios using computer simulations.

3 DDDAS Architecture

The integrated adaptive cyberinfrastructure will contain several components that communicate with each other (Fig. 1). These components can be abstracted at the highest level into three categories: sensors & data, algorithms & models, and middleware & resources. These components will be assembled and driven by an adaptive workflow engine and a web portal. Each component is described below.

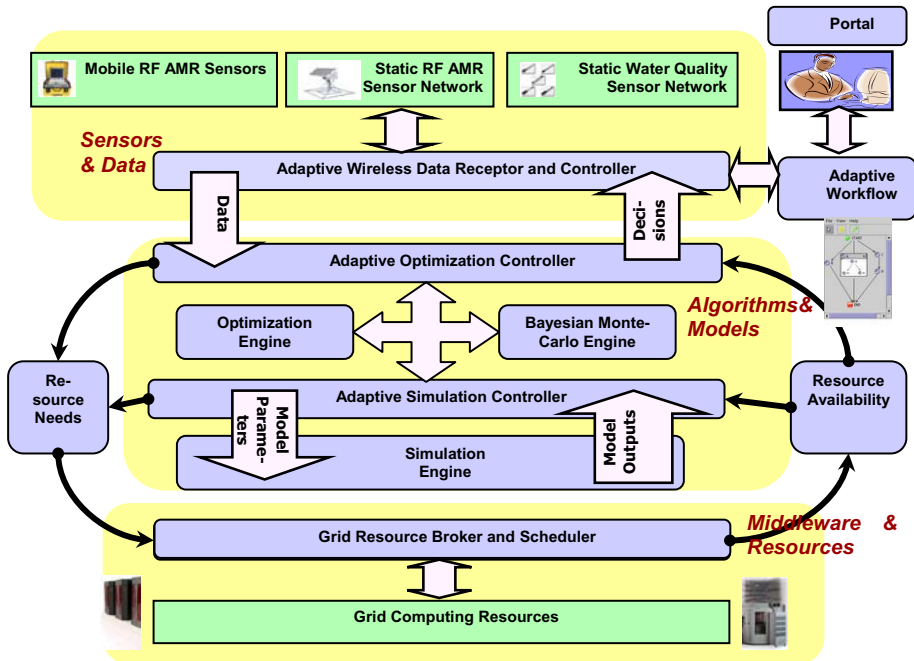


Fig. 1. Basic Architecture of the Cyberinfrastructure

3.1 Sensors and Data

Part of the infrastructure relies on the continuous ingestion of sensor data to drive the simulations. In case of an incident, it is planned to run several alternative instantiations in parallel and present the observing team with statistical augmentations in relationship to their accuracy. Our framework will furthermore determine areas of interest and suggestions for locating additional sensors. The sensor data is gathered from sensors supplied by the Neptune Technology Group (NTG). These second generation sensors can be integrated into a wireless RF (Radio Frequency) network. The data can either be read by drive-by readings or a communication station placed in the vicinity of a number of fixed transmitters. The sensors are one-way communication devices that transmit data using frequency hopping spread-spectrum technology enabling data security and reliability. The frequency for data reporting of the sensors is between 11 and 14 seconds. The fixed transmitters are configured to report 15 measurements per

hour back to the utility center. The data gathered at the data center will be one way forwarded to a specialized and secured data store that is used by our software in order to avoid interference with the sensor network.

Adaptive Data Receptor and Controller. We will construct a stochastic network simulation and data communication engine to replace the physical radio frequency data communication layer. Working with NTG, their EZNET™ server software will be modified to communicate directly with a defined software API that simulates the physical behavior of installed RF meter interface units. At the heart of this new capability is simulation of the NTG E-Coder™ and ProRead™ encoder meter register, which has a physical interface with the customer water meter. In the simulation environment, these meter registers will be populated by network simulation models driven by stochastic models of individual customer water usage. The stochastic simulation models will be based on an existing framework for Monte-Carlo simulation of network hydraulics and water quality [3] and published Poisson-process stochastic water usage models (e.g.,[4]). The general testing environment will also simulate – as it would be configured in a real-time installation – contaminant introduction and the response of installed water quality sensors and data communications, thus providing a flexible testbed for evaluating the performance of our cyberinfrastructure.

3.2 Models and Algorithms

Our cyberinfrastructure is designed to tackle two kinds of problems: (i) contaminant source characterization problem, (ii) contaminant control problem. Immediate focus within this project will be on the source characterization problem which can be posed as parameter estimation or an inverse problem. The goal is to use the time-series of sensor data to recover the likely locations of the contaminant sources and their release histories that minimize the error between the predictions and the observations. The control problem, to be considered in subsequent projects, can be posed as a strategy optimization problem where the goal is to identify an optimal schedule for activating a set of control choices to meet threat management objectives. Choices for controlling the hydraulics in the network include pumping rates, valve adjustments, and flow releases from fire hydrants.

Optimization Engine. The optimization engine will consist of a suite of optimization procedures that will be used to search for the most likely source characterization to fit the measurements, calibrate the model in real time as new data are streamed-in, and to search for the most effective control strategy to manage network contamination. To facilitate use of these methods within the adaptive cyberinfrastructure, methods with varying degree of sophistication are needed to appropriately match the data and computing resource availability as well as the required accuracy at different stages of solution. The search methods will incorporate incoming data from sensors as they are streamed in real time to improve prediction efficiency. We are currently investigating an adaptive evolutionary algorithm (EA) for conducting search under dynamically varying systems [5]. The promising preliminary results indicate that this approach significantly improves prediction efficiency with dynamic infusion of data. An

associated investigation of another search procedure shows promise in assessing the degree of non-uniqueness in solutions as more data are incorporated into the prediction process [6].

Bayesian/Monte-Carlo Engine. The Bayesian/Monte Carlo engine will consist of procedures to identify adaptive and cost & time-efficient sampling plans (locations and frequencies) to reduce model-based prediction uncertainties and therefore improve confidence in control strategy performance. A computing resource-adaptive Bayesian data worth framework will be implemented. A data worth framework is a sequential process where in each iteration the best location to next sample is identified - sampling at that location will maximize the expected worth of the sample. The incorporation of Bayesian analysis allows the accounting, and updating with new sampling data, of uncertainties in parameters that are not directly observable, e.g., the true location and injection rates of contaminants in the water distribution network. Markov chain Monte Carlo (MCMC) methods for Bayesian analysis [7] will be used for computational efficiency. We will research and implement MCMC methods that can adapt to resource fluctuations in the grid computing environment: the number of parallel MCMC chains and tolerances for stopping criterion will be controlled. Approaches for reconciling the tradeoff between accuracy and run time will be explored. The accuracy in the identification of the best next sampling location will be balanced against the need to quickly zero in on the contaminant source characterization.

Simulation Engine. EPANET [8], a widely used water distribution network hydraulic and water quality modeling tool, will be used as the simulation engine. This model uses known pipe network topology, link/node physical characteristics, and network boundary and initial conditions, to simulate the space-time variation of flows, pressures, and water quality concentrations using well-established principles [9]. The EPANET engine is available as a C language library with a well-defined API [10]. EPANET will be made malleable through adaptive control by the simulation controller. Malleability features to be incorporated are: (i) ability to change the problem configuration at any given time step to meet resource limitations or problem requirements, (ii) ability to change input conditions as new data become available, and (iii) ability to communicate with the controller at any given time. For (i), sensed information and the current solution state would be used to adaptively update the problem configuration. Implementing (ii) is straightforward as new sensor data will replace old values used in the simulation. For (iii), a communication feature using XML-based message passing will be built into EPANET so that it can readily communicate with controller in sending and receiving metadata augmented messages. The messages will include input data (e.g., decision variables) as well as action flags that can steer the simulation. The simulation engine will be coupled with an existing test suite of five real network models, ranging in size from several hundred to over 12,000 nodes. The test suite represents a range of hydraulic behaviors. Thus the test networks will adequately support development of algorithms, as well as challenging their performance and scalability.

Optimization and Simulation Controllers. The optimization and Monte-Carlo controller will have the ability to (i) select appropriate optimization algorithms or Monte-Carlo

simulation depending on the problem scenario, data, or computer resources, (ii) communicate with the optimization and Monte-Carlo engines, (iii) communicate with the adaptive data controller, (iv) communicate with the resource broker, and (v) communicate with the global workflow portal/controller. For (i), a set of predefined rules based on solution time and problem scenario will be formulated for different algorithms. These rules will then be used in an adaptive manner to select the appropriate algorithm.

The simulation controller will have capabilities to (i) communicate with the global workflow controller, (ii) communicate with the optimization and Bayesian/Monte-Carlo engines, (iii) adaptively change model data, (iv) steer and track the model simulations, (v) communicate with the resource broker. The global workflow controller will relay the baseline model scenario to the model controller. The optimization and Bayesian engines will relay the number of model evaluations to be completed, the decision variables, and any additional data from the sensors to the model controller. The resource broker will relay the available computer resources to the model controller. The model controller will utilize this information to determine the optimal model resolution and the resource requirements. It will then communicate with the resource broker and launch the appropriate simulations. During the simulations, the model controller may change the model data, launch additional simulations, or terminate the simulations depending on the need.

3.3 Middleware and Resources

The requirement to manage dynamic processes as part of our implementation strategy calls for a framework that assists the orchestration and coordination of these processes within a dynamically changing infrastructure. One part of such a framework is the design of a workflow management system [11,12] that provides interfaces among the sensor network, the scientific model calculations, the available Grid infrastructure, and the human participants. Such an integrated workflow system allows us to plan and organize the complex experiments in a dynamic changing environment. In contrast to other systems, we project an integrated approach that not only allows us to extend our system, but also allows us to integrate with other systems through the introduction of convenient abstractions. One of the open research issues in the current development of workflow systems is: How do we build a supporting workflow system that not only maps the workflow to an existing static or dynamic environment, but also modifies the workflow at runtime to adapt the workflow itself to the changing environment? This is particularly challenging as we foresee scenarios in which we can for example change the accuracy of calculations performed as part of the workflow while reducing the resource requirements in time and space.

The specification of such workflows must be assisted through a number of frameworks. We need workflow shells for rapid prototyping, a graphical editor to assemble the components by the application user, a programming framework to enhance the workflow management system, and an XML-based specification to assist in interfacing with the Web services community (a language such as XML is intended for automatic interpretation and is too cumbersome for humans to read). We will address these issues by expanding the Java CoG Kit [12] workflow system and explicitly focusing on the needs of this user community.

Grid resource broker and scheduler. Our use-cases include two scenarios, a traditional scenario in which we attempt to access a number of resources available to us for running a calculation, and another in which we attempt to predict a very accurate calculation in a case of an emergency. In the first case, we will use facilities of existing Grid middleware, such as Globus Toolkit [13], Condor glide-ins, and the Java CoG Kit to fulfill the users' need for accessing the widest range of resources. We also need to access, however, an advanced reservation system that allows preemptive scheduling based on priority needs in case of an emergency. To achieve this goal we plan to engage with the TeraGrid community in order to evaluate an on-demand emergency ticket system.

Adaptive workflow portal. One important issue is to enable simple access by the user community through a portal. We will reuse and expand upon portal technology developed as part of the NMI OGCE project [14]. We will enhance this workflow portal with the ability to browse through dynamically changing workflows. We focus on the development of components that foster ease of use, reusability, and computational steering within our framework. A workflow component repository that we have proposed recently [11] will be implemented to allow us to dynamically change components at runtime and develop automatic adaptive algorithms.

4 Conclusion

We have presented a novel infrastructure for dynamic data driven analysis for threat management in urban water distribution systems. We believe such an architecture can enable urban authorities to effectively manage water distribution contamination incidents by incorporating and controlling real-time feed of data from automatic meter readers and sensors.

Acknowledgements

This work is supported by National Science Foundation (NSF) under Grant Nos. CMS-0540316, ANI-0540076, CMS-0540289, CMS-0540177, and NMI-0330545. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The Java CogKit development is supported by NSF grant nos. ANI-0540076, NMI-0330545, and by the Dept of Energy under contract no. W-31-109-Eng-38.

References

1. Water Science and Technology Board, A Review of the EPA Water Security Research and Technical Support Action Plan: Parts I and II, *The National Academies Press*, 2003.
2. Darema, F. Dynamic data driven applications systems: New capabilities for application simulations and measurements. *Lecture Notes in Computer Science 3515*, pages 610–615, 2005.

3. Murray, R., R. Janke, and J. Uber. The threat ensemble vulnerability assessment (TEVA) program for drinking water distribution system security. In Proceedings, *ASCE/EWRI World Water and Environmental Resources Congress*, Salt Lake City, UT, 2004
4. Buchberger, S., and G. Wells, "Intensity, Duration, and Frequency of Residential Water Demands," *J. Water Resour. Plng. and Mgmt.* 122(11), 1996.
5. Liu, L. and S. R. Ranjithan, Adaptive Niche Co-Evolutionary Algorithm (ANCEA) for Dynamic Optimization, submitted to: *Genetic and Evolutionary Computation Conference (GECCO) 2006*, ACM.
6. Zechman, E. M. and S. R. Ranjithan, Niche Co-Evolution Strategies to Address Non-uniqueness in Engineering Design, submitted to: *Genetic and Evolutionary Computation Conference (GECCO) 2006*, ACM.
7. Gilks, W.R., S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*, chapter Introducing Markov Chain Monte Carlo, pages 1–20. Chapman & Hall, London, 1996.
8. Rossman, L.A., *EPANET Users Manual*. Risk Reduction Engrg. Lab., U.S. Environ. Protection Agency, Cincinnati, Ohio, 1994.
9. Walski, T., *Analysis of Water Distribution Systems*, Van Nostrand Reinhold, New York, 1984.
10. Rossman, L.A.. The epanet programmer's toolkit. In Proceedings of Water Resources Planning and Management Division Annual Specialty Conference, ASCE, Tempe, AZ, 1999.
11. von Laszewski, G. and Kodeboyina, D. A Repository Service for Grid Workflow Components. *International Conference on Autonomic and Autonomous Systems*. IEEE, 23-28 October 2005. <http://www.mcs.anl.gov/~gregor/papers/vonLaszewski-workflow-repository.pdf>
12. von Laszewski, G. and Hategan, M. Java CoG Kit Workflow Concepts, in *Journal of Grid Computing*, Jan. 2006. <http://www.mcs.anl.gov/~gregor/papers/vonLaszewski-workflow-jgc.pdf>.
13. The Globus Project. Web Page. Available from: <http://www.globus.org>.
14. OGCE. Open Grid Computing Environments. Web Page. Available from: <http://www.ogce.org>.

Model-Driven Dynamic Control of Embedded Wireless Sensor Networks

Paul G. Flikkema¹, Pankaj K. Agarwal², James S. Clark², Carla Ellis²,
Alan Gelfand², Kamesh Munagala², and Jun Yang²

¹ Northern Arizona University, Flagstaff AZ 86001 USA

² Duke University, Durham, NC USA

Abstract. Next-generation wireless sensor networks may revolutionize understanding of environmental change by assimilating heterogeneous data, assessing the relative value and costs of data collection, and scheduling activities accordingly. Thus, they are dynamic, data-driven distributed systems that integrate sensing with modeling and prediction in an adaptive framework. Integration of a range of technologies will allow estimation of the value of future data in terms of its contribution to understanding and cost. This balance is especially important for environmental data, where sampling intervals will range from meters and seconds to landscapes and years. In this paper, we first describe a general framework for dynamic data-driven wireless network control that combines modeling of the sensor network and its embedding environment, both in and out of the network. We then describe a range of challenges that must be addressed, and an integrated suite of solutions for the design of dynamic sensor networks.

1 Introduction

The technology of wireless sensor networks is now becoming a mature research field. As a result, the discipline is undergoing a natural evolution into cross-cutting subtopics and applications, and the original vision of smart dust—myriads of tiny, minimalist sensors—is now just one niche. Given the physical reality of many applications, especially the size and expense of physical transducers and the cost of deployment, we are now seeing a movement toward networks that are comparatively lower in population and density. However, integral to this paradigm is the notion that the sensor nodes should be much smarter. We will argue that a key aspect of this intelligence is that these sensors are, surprisingly, even *more embedded* into their environment. This tight coupling between the sensed and the sensors results from the need for the network to have adequate explanatory power under significant energy consumption constraints.

Our particular application is the monitoring of physical environments and ecosystems, with the goal of predicting biodiversity and carbon accumulation based on scenarios of change in the atmosphere and disturbance. Changing temperature, moisture availability, and atmospheric CO₂ concentrations determine the rates at which leaves gain carbon and, thus, the rates of tree growth. While it is clear that species respond differently to climate and CO₂, heterogeneous data,

combined with modeling challenges, have frustrated efforts to scale such relationships in ways that would permit reasonable estimates of impacts on biodiversity (see e.g. Clark et al. 2001).

We believe that *dynamic* wireless sensor networks promise to revolutionize understanding of such complex phenomena by using statistically-aware dynamical models that guide their responses to changing sampling and communication needs in the context of evolving data acquisition and communication costs. The underlying simulation models are computationally intensive and ingest data with disparate natural scales. Current wireless networks are capable of meeting many of the data needs, but network control algorithms must be developed to steer sampling and collaboration based on complex multi-level models. By injecting state-of-the art dynamical models and statistical computation into the network, complemented by out-of-network modeling, the network can acquire data at relevant scales and drive models that aid understanding and anticipate change.

2 Dynamic Sensor Networks

If sensor networks are to revolutionize ecological forecasting, they must become autonomous instruments that employ dynamic model-driven control of the sensing process, including sampling, communication, estimation, prediction, and model inference. A framework that encompasses these ideas must also accommodate the limited resources available in the network, complementing them with out-of-network capability. Figure 1 depicts such a framework, showing the two key functions, modeling and control, distributed between the network and out-of-network (i.e., in the lab) computational resources. The elements are in-Network Inference and Prediction (NIP) for real-time assessment of the system, Dynamic In-Network Control (DINC) for real-time adaptation, and Dynamic Out-of-Network Control (DONC) for network analysis, adaptive deployment (re-configuration), and trouble shooting. To manage complexity, computationally

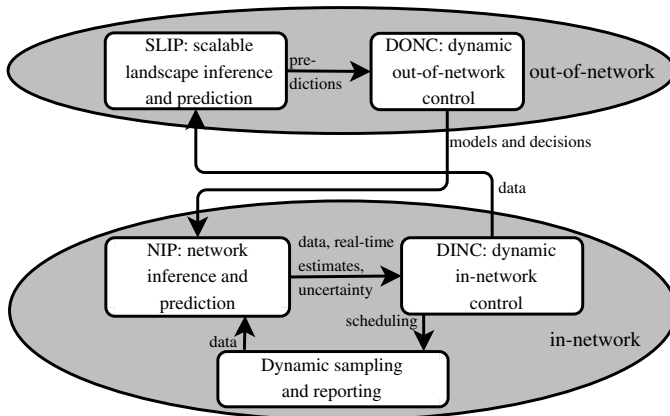


Fig. 1. Components of network control

demanding and long-term tasks are handled by a Scalable Landscape Inference and Prediction (SLIP) model operating outside the network along with DONC processing.

The SLIP model of forest stand dynamics focuses on 1) data assimilation from environmental variables, tree growth, and tree demography and 2) algorithm development to permit efficient computation for change over large landscapes. Data range from long-term (up to 15 yr) experiments involving manipulating canopy characteristics, CO₂ (FACE experiments), herbivores, remote sensing, and wireless networks of environmental sensors. Models involve mass (water, CO₂), and energy exchange, including basic tree physiology, and growth dynamics of trees. To simulate large landscapes, the SLIP model makes use of recent developments in algorithms, data structures, and hardware (Govindarajan et al. 2004, 2005). The SLIP model is also the basis for in-network NIP modeling (Figure 1) and as input to DINC and DONC.

In numerous applications, complex models will actually be networks constructed of component models. Both within and outside the *in situ* network, we expect that these component models will execute in real time with varying degrees of collaboration and competition. Given that communication is dramatically more expensive than computation in wireless networks, in-net and out-of-net model designs will be strongly influenced more by comparative communication costs than computational constraints. Consequently, in-network models will rely heavily on local, inexpensive information, informed by out-of-net models that fully exploit evolving global datasets.

A critical aspect of both the SLIP and NIP models is the integration of estimation and prediction within a Bayesian inference framework. The advantage is predictive intervals that are directly anchored by data. Hierarchical Bayes modeling (see e.g. Smith and Gelfand 1992) has emerged as a powerful new tool for inference, prediction, and decision in high-dimensional systems, and has been further developed for environmental applications (Agarwal et al. 2002) and specifically for modeling tree stand dynamics (Clark 2005). The approach provides a basis for integrating the extensive data from a range of scales. It allows modeling a complex system as a network of relationships among elements, both known and unknown. If we have set of parameters R specifying a particular process model Φ , data assimilation within the hierarchical Bayes framework is based on the decomposition

$$[\text{data}, \Phi, R] = [\text{data}|\Phi, R][\Phi|R][R].$$

Each of the three conditional relationships might be further decomposed, depending on complexity. Data can enter into different stages of the model, with each data type related to one or more elements of the process tailored to the appropriate scale, measurement errors, and so on. Parametric modeling at the various stages provides for context, which may be spatial, temporal, or involve relationships among subsets of observations.

In the context of this research, the SLIP simulator serves as the out of network model (Figure 1). It makes use of sophisticated algorithms to capture large

numbers of interactions across broad landscapes in reasonable time. Because of computational and memory demands, SLIP will run on a server or cluster outside the wireless network itself. DINC responds to NIP predictions in real time, with updated scheduling based on optimization. NIP also passes information to Dynamic Out-of-Network Control (DONC), which provides for more detailed assessment and control.

3 Sensor Network Design and Implementation

Wireless sensor network architectures and implementations should reflect the specific opportunities and constraints of the environmental sensing application. First, research-quality scientific data requires external probes, flexible and quiet analog circuit design, and a weatherproof package. Secondly, for scientific-quality sensing applications, large-scale redundancy (to aid battery lifetime or allow disposability) is economically infeasible. For example, the cost of probes is approximately 40% of the total, so that the networking and sensing infrastructure for a single node can exceed \$1,000 at the prototype stage. Moreover, installation costs are significant due to the careful probe deployment procedures required. Hence optimization of spatial sampling and model-aided redeployment, in addition to dynamic temporal sampling and reporting, will be a growing concern in ecosystems sensing applications.

The prototype WiSARDNet (wireless sensing and relay device network) sensor network technology (Yang 2005, www.wisardnet.nau.edu) targets applications in biological and ecological field research. A WiSARDNet consists of two classes of nodes; 1) many nodes that provide sensing and multihop networking capabilities, and 2) relatively few gateways to the internet for SLIP and DONC functions. WiSARD nodes use a dual-processor design, with the labor divided between a brains board that provides communication and networking services and a probe data acquisition board that handles the details of the sensing tasks. The hardware design is a three-board stack; the data acquisition board used in WiSARDs is replaced in gateways by a board that provides communication interfaces and non-volatile memory for data archival.

The WiSARDNet protocol stack design is driven by the sensing application, wherein communication can be scheduled because of the low temporal sampling rates for environmental phenomena. These rates are not necessarily static: adapting sampling to hot or cold spots in time or space allow for improved data quality at a fixed average sampling rate. The protocol stack is based on proactive coordination, or scheduling, of communication transactions. We have implemented a media access control (MAC) algorithm (Flikkema 2003) that employs local proactive coordination in a slotted-time framework so that a node is awake—and consuming energy—only when it has scheduled communication with one of its cliques (sets of neighboring nodes). The MAC layer is coupled with lower and higher layers in the service of energy efficiency, scalability, and robustness.

Dynamic sensor node design. Most current wireless sensor nodes target minimum energy consumption and 8-bit processors that use little power but have

very limited computational capability. The most obvious requirement driven by dynamic sensing is a hardware/software design that is capable of significantly higher computational power while maintaining low energy consumption. Next-generation 16- and 32-bit embedded processors are being introduced that provide order-of-magnitude increases in computational throughput over the 8-bit microcontrollers while using far less energy than 32-bit architectures now used in easily rechargeable devices such as PDAs. New devices provide for energy management with programmable subsystem clocks and a low-power instruction cache. On-chip memory management units allow separate logical program and data spaces, enabling protection critical in systems with dynamic software re-configuration.

Support for over-the-network dynamic reconfiguration is also required, including loading of new tasks and programs and replacement of existing components. Progress in this direction includes the module registration approach in the SOS operating system (Han 2005) to provide protection from calls to nonexistent or outdated components. Efficient implementation of code and data protection can use hardware memory management.

Sensor network OS's should be extended to include application-level support of energy-consumption management. The Energy Centric Operating System (ECOSystem) (Zeng et al. 2002, 2003) has demonstrated how to schedule computation and communication to satisfy a specified power budget to guarantee a specified battery lifetime.

4 Algorithm and Software Development

Data service layer. Casting model-driven adaptive sensing and reporting as a data management problem may ultimately broaden the functionality of sensor networks (overview in Kumar 2003). We are currently building a *data service layer* for WiSARDNet that supports a high-level programming interface for specifying data acquisition tasks. These tasks may acquire information about both the environment as well as the system itself (e.g., remaining battery power and storage capacity at each sensor). High-level specifications of tasks simplify programming and enable greater optimization opportunities.

The data service layer is jointly implemented by a central server and the nodes in the WiSARDNet. Given a sensing task, the central server carries out optimization based on models with previously assimilated data, and obtains an initial acquisition plan to be deployed in the network. This plan consists of sub-plans to be executed at individual nodes, which specify when and what to sample and report. The fundamental building blocks of plans are *dynamic continuous queries*: They are continuous in that they continuously execute over streaming input data and produce results; they are dynamic in that they admit dynamic external control—these two features are essential in supporting NIP, DINC, and DONC. Together, the dynamic continuous queries form data/control flows over the network, where one query's output may be another query's data or control input. Each node in the WiSARDNet runs a data service component

that interprets the deployed plan and executes the local sub-plans. The data service component coordinates the execution and adaptation of sub-plans and their interaction with WiSARDNet hardware probes and communication layer, tags data reports with meta-data about sampling and measurement conditions, and provides multi-resolution data storage within the network.

Model-driven sensing and reporting. Environmental and system modeling offers many opportunities for optimization. With the out-of-network models of ecological processes (SLIP), we can quantify the utility or fidelity of data (in both accuracy and timeliness) and use it to control sampling and reporting in the WiSARDNet (DONC). As noted earlier, the utility of an observation depends on how it is to be used. For example, even if the local estimate of soil moisture is precise enough for water balance modeling, a model for plant mortality risk may require a more precise estimate. Thus, DONC must consider the requirements and needs of multiple models.

Inside the network, as new sensor readings are acquired, dynamic continuous queries running on the nodes collectively maintain local models of the environment (NIP). Within the fidelity requirements set by DONC, we use the in-network models to control sampling and reporting adaptively (DINC) in order to achieve energy savings. In particular, we may choose not to acquire a sensor reading if, according to DONC, its utility is low or its value can be confidently predicted by NIP to within the required accuracy. Furthermore, instead of reporting every raw sensor reading to the central server, the network only needs to report updates to local model parameters or deviations of readings from local model predictions.

Finally, our system also maintains reliability models of sensors and communication links, and uses them with statistical inference techniques to reason about errors and failures, which routinely occur in wireless sensor networks. Models of battery lifetime provide us with the basis for energy-based optimization.

Algorithmic challenges. In order to fully realize the potential of the model-driven, dynamic-system approach to understand complex environmental phenomena, one has to address the algorithmic challenges at various levels. One of the basic challenges is to define the right utility and fidelity measures based on the underlying models, which can provide the framework for developing clever algorithms that maximize fidelity with as little resources as possible. These measures will define the (often conflicting) optimization criteria, guide various trade offs, etc.

One of the main algorithm challenges is to develop algorithms that provide various trade-offs in a dynamic sensor network, including: fidelity vs. accuracy, data fidelity vs. sensing and reporting cost; model maintenance cost vs. cost-saving potential; cost vs. benefit of dynamic adaptation; and cost vs. benefit of failure protection. We have recently begun research in this area. Silberstein, Braynard, and Yang (2006) investigate the trade-off between a purely temporal model and a spatio-temporal model for suppressing change reporting in a sensor network; the former only exploits temporal coherence while latter exploits both temporal and spatial coherence. We are studying similar trade-offs for the

forest growth model (Chakraborty, et al., 2006). In the same vein, there has been preliminary work (Silberstein, Munagala, and Yang 2006) on developing power-efficient distributed local control algorithms to exploit spatio-temporal correlations to perform query-specific model-driven data acquisition. Some recent work (Goel, Guha, and Munagala 2006; Flikkema 2006) presents a theoretical framework, algorithms (both adaptive and non-adaptive), and analysis of the trade-offs between acquisition cost and information gained in model-driven resolution of uncertain data.

A related challenge is to develop an optimization framework that spans across multiple software layers and executes both inside and outside the network. Each software layer—application, data service, OS, and networking—continuously carries out optimization and adaptation at runtime, and a coordination among them is needed to maximize the performance. Similarly, DINC and DONC must coordinate with each other to ensure overall efficiency and stability of the system. Furthermore, optimization could be expensive. For example, when optimizing sensor data acquisition tasks, reasoning with complex statistical models can be a challenging computational task, especially when complicated by energy costs and network topology. Our recent work (Silberstein, et al. 2006) involves developing computationally efficient techniques based on sampling and linear programming with both energy and network considerations.

Finally, the availability of sensor data provides many opportunities to significantly improve environmental modeling. For example, one can run environmental models at a finer time scale and thus incorporate variations in the values of parameters (instead of drawing them from a distribution). But these opportunities cannot be realized without addressing a number of algorithmic challenges in the simulation of complex models. For example, one may have to use an event based simulation (e.g., using kinetic data structures (Agarwal et al. 2002)) and update the information locally and only when it is necessary. In some of our previous work, we have exploited spatial coherence but did not pay enough attention to temporal coherence (Govindrajan et al. 2004, 2006). We are currently developing algorithms for the SLIP model that exploits both temporal and spatial coherence.

5 Conclusion

One of our goals is to experimentally evaluate whether a network can deliver better results than the standard fixed sampling and out-of-network modeling in the context of changing real-world environmental conditions. Ultimately, results from these experiments will inform the design of an accurate, dynamic, energy-efficient production networks tailored to the specific demands of the application.

Our effort to understand the biodiversity and carbon consequences of environmental change is broad enough to encompass many of the types of challenges faced by dynamic data-driven application systems. The research results will be applicable to many environmental sensing/decision applications, including micrometeorological sensing, pollution monitoring and environmental remediation,

and public security/safety. In particular, results from this effort will inform the planning and implementation of future distributed in situ monitoring, modeling, and decision systems.

References

1. Agarwal, P. K., Guibas, L. J., et al: Algorithmic issues in modeling motion, *ACM Comput. Surv.* 24 (2002), 550–572.
2. S. Chakraborty, P. K. Agarwal, and J. Clark, The gap light model, manuscript, 2006.
3. Clark, J.S., et al: Ecological forecasts: an emerging imperative, *Science* 293:657-660, 2001.
4. Clark, J.S.: Why environmental scientists are becoming Bayesians, *Ecol. Lett.* 8:2-14, 2005.
5. Flikkema, P., West, B.: Clique-Based Randomised Multiple Access for Energy-Efficient Wireless Ad Hoc Networks, in *Proc. 2003 IEEE Wireless Communications and Networking Conference (WCNC '03)*, New Orleans, March 2003.
6. Flikkema, P: The precision and energetic cost of snapshot estimates in wireless sensor networks. Submitted for publication (2006)
7. Goel, A., Guha, S., and Munagala, K.: Asking the right questions: Model-driven optimization using probes. Submitted for publication (2006)
8. Govindrajan, S., Dietze, M., Agarwal, P. K., and Clark, J., A scalable simulator for forest dynamics, in *Proc. 20th Sympos. Comput. Geom.* (2004).
9. Govindrajan, S., Dietze, M., Agarwal, P. K., and Clark, J., A scalable algorithm for dispersing population, *J. Intelligent Information Systems*, in press.
10. Kumar, V. ed.: Special Section on Sensor Network Technology and Sensor Data Management (Part I). *SIGMOD Record*, **32**(4) (2003)
11. Silberstein, A., Braynard, R., Ellis, C., Munagala, K., and Yang, J.: A sampling-based approach to optimizing top-k queries in sensor networks. *Proc. of the 22nd Intl. Conf. on Data Engineering*, Atlanta, Georgia (2006)
12. Silberstein, A., Braynard, R., and Yang, J.: Constraint-chaining: on energy-efficient continuous monitoring in sensor networks. *Proc. of the 22nd Intl. Conf. on Data Engineering*, Atlanta, Georgia (2006)
13. Silberstein, A., Munagala, K., and Yang, J.: Energy efficient monitoring of extreme values in sensor networks. Submitted for publication (2006)
14. Smith, A.F.M., and Gelfand, A.E.: Bayesian statistics without tears: a sampling-resampling perspective. *American Statistician*, 46, 84-88, 1992.
15. Yang, Z., et al.: WiSARDNet: A system solution for high performance in situ environmental monitoring, 2nd International Workshop on Networked Sensor Systems (INSS 2005), San Diego, 2005.
16. Heng Zeng, Carla S. Ellis, Alvin R. Lebeck, and Amin Vahdat. Ecosystem: Managing energy as a first class operating system resource. In *Proc. Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)*, pages 123–132, October 2002.
17. Heng Zeng, Carla S. Ellis, Alvin R. Lebeck, and Amin Vahdat. Currency: A unifying abstraction for expressing energy. In *Usenix Annual Technical Conference*, pages 43–56, June 2003.

WIPER: The Integrated Wireless Phone Based Emergency Response System*

Gregory R. Madey¹, Gabor Szabo², and Albert-László Barabási²

¹ Computer Science & Engineering

University of Notre Dame

Notre Dame, IN 46556, USA

gmadey@nd.edu

<http://www.nd.edu/~gmadey/>

² Department of Physics

University of Notre Dame

Notre Dame, IN. 46556, USA

{gabor.szabo, Albert.L.Barabasi.1}@nd.edu

<http://www.nd.edu/~alb>

Abstract. We describe a prototype emergency response system. This dynamic data driven application system (DDDAS) uses wireless call data, including call volume, who calls whom, call duration, services in use, and cell phone location information. Since all cell phones (that are powered on) maintain contact with one or more local cell towers, location data about each phone is updated periodically and available throughout the cellular phone network. This permits the cell phones of a city to serve as an ad hoc mobile sensor net, measuring the movement and calling patterns of the population. Social network theory and statistical analysis on normal call activity and call locations establish a baseline. A detection and alert system monitors streaming summary cell phone call data. Abnormal call patterns or population movements trigger a simulation and prediction system. Hypotheses about the anomaly are generated by a rule-based system, each initiating an agent-based simulation. Automated dynamic validation of the simulations against incoming streaming data is used to test each hypothesis. A validated simulation is used to predict the evolution of the anomaly and made available to an emergency response decision support system.

1 Introduction

During a disaster, emergency response managers could benefit from timely alerts and quality information about the location and movement of the entire affected population. Reports from on-scene coordinators, first responders, public safety officials, the news media, and the affected population can provide managers with point data about an emergency, but those on-scene reports are often inaccurate, conflicting and incomplete with gaps in geographical and temporal coverage. Additionally, those reports must be merged into a coherent evolving picture of the entire affected area to enable emergency managers to effectively respond.

* The material presented in this paper is based in part upon work supported by the National Science Foundation, the DDDAS Program, under Grant No. CNS-050312.

Mobile phones are becoming increasingly ubiquitous throughout large portions of the world, especially in highly populated urban areas. Mobile phone providers have available real-time data about the call volume, calling patterns, and the location of the cell phones of their subscribers. In order for a cell phone to place outgoing calls and to receive incoming calls, it must periodically report its presence to nearby cell towers, thus registering its spatial-temporal position in the geographical cell covered by one of the towers. Thus, in addition to the call volume and call patterns of their subscribers, cellular carriers can provide data about the collective location and movement of all the cell phones (those powered-on) in an area affected by a disaster.

We describe the design and development of a prototype emergency response system that would use streaming data from the cellular carriers in an affected area to provide alerts, interpretation, and predictions of the evolution of the emergency event. Recent events in New Orleans and Houston (USA) with hurricanes Katrina and Rita motivate the need for such a system. Other disasters caused by earthquakes, floods, tsunami, tornados, terrorist attacks, industrial accidents, and civil disorders also suggest applications for the system. Included in the design is an agent-based simulation system that dynamically requests detailed data about cell phone activity in the area modeled, including triangulated data with more precise location information. The simulation system uses the streaming data to calibrate and dynamically validate one or more agent-based simulations of the emergency. Validated simulations can then be used to predict the evolution of the emergency and help emergency managers to anticipate events and needs of the affected population.

The design of the prototype Wireless Phone Based Emergency Response System (WIPER) is inspired by the concepts of Dynamic Data Driven Application Systems (DDDAS), which is briefly summarized in the next section. The subsequent section describes the WIPER design, including 1) the *system architecture* – application of open-source/open-standards Service Oriented Architecture (SOA) middleware tools, grid services, software and servers used to implement the modular distributed WIPER prototype, 2) the *dynamic data driven simulations* – knowledge-based & agent-based simulation system used to classify, test, and predict the course of an anomaly, and 3) the *web services-based decision support system* – the end-user composition of web services based workflow for a flexible web-based decision support system. We conclude with a summary and discussion of the limitations and potential privacy challenges to deployment of the WIPER system.

2 Dynamic Data Driven Application Systems

The concept of dynamic data driven application systems (DDDAS) was first explored in detail in a NSF workshop in early 2000 [1]. That workshop concluded that the DDDAS concept (i.e., that simulations and real-world data be merged into symbiotic feedback control systems) offered the potential of greatly improving the accuracy and efficiency of models and simulations. The workshop final report recommended more research in the areas of 1) dynamic, data driven application technologies, 2) adaptive algorithms for injecting and steering real-time data into running simulations, and 3) systems software that supports applications in dynamic environments. At following conferences and workshops, initial research and applications exploring these research

areas were reported [2]. A fourth area of research important to the DDDAS concept emerged, that of measurement systems; the dynamic steering of the data collection needed by the simulations may require improvements in measurement, observation and instrumentation methods. In 2004, Darema described the DDDAS concept as:

Dynamic Data Driven Application Systems (DDDAS) entails the ability to incorporate additional data into an executing application - these data can be archival or collected on-line; and in reverse, the ability of applications to dynamically steer the measurement process. The paradigm offers the promise of improving modeling methods, and augmenting the analysis and prediction capabilities of application simulations and the effectiveness of measurement systems. This presents the potential to transform the way science and engineering are done, and induce a major impact in the way many functions in our society are conducted, such as manufacturing, commerce, hazard management, and medicine [2].

Several workshops and symposia focused on the DDDAS concept have been held with proceedings available online [3].

The prototype WIPER system explores all four research areas relevant to the DDDAS concept: 1) it dynamically responds to streaming real-time data and steers the collection process, 2) it includes both algorithms for detecting anomalies that may indicate an emergency and a rule and agent based system that dynamically incorporates new data into its analysis, 3) it employs the recently developed open standards based service oriented architecture (SOA) for integrating the multiple distributed modules that comprise the WIPER system, and 4) the data that streams into the system from the cellular carriers is dynamically focused on the locality of the emergency, detailed data rather than aggregate data is collected, and when needed by the simulation system, higher precision triangulated data is requested from the carriers.

3 Data

The design and development of the WIPER system uses both actual call and location data provided by a cellular carrier and synthetic data to simulate emergencies. All data is anonymized to protect privacy. During development, testing and evaluation the data is stored in a database with several modules streaming the data to simulate the real-time data streams the system would see if deployed. The data that the WIPER system uses is aggregate in nature, does not track individual cell phones by actual user ID, and does not include the content of phone calls or messages. A discussion of privacy issues is included in the final section of this paper.

4 Design

The design and operation of the WIPER DDDAS prototype is shown schematically in Fig. 1. The system has three layers: 1) Data Source and Measurement Layer; 2) Detection, Simulation and Prediction Layer; and 3) Decision Support (DSS) Layer. Each is described briefly in the following subsections, along with details on the agent-based simulations and the system architecture used by the WIPER system.

4.1 Data Source and Measurement Layer

For the development, testing and evaluation phases of the WIPER system, the Data Source and Measurement Layer contains both real historic data and synthetic data stored in three online database systems: RTDS, HIS and TRI. The *Real Time Data Source* (RTDS) contains the full activity and spatial location data automatically collected by the mobile phone company, including the 30 sec CDR (Call Detail Record) tags on all phones, identifying the closest cell tower to the phone, and activity data, such as the initiator and the recipient of a call, its duration and the nature of service used. The *Historical Data* (HIS) is generated from RTDS, and stores aggregated reference information used to train and calibrate the system that detects communication and spatial anomalies. Finally, the *Triangulation Application* (TRI) is capable of providing high precision location information on selected phones. It is activated only on request, and is steered by either the SPS or DSS modules. The interface to the three data sources is designed such that real-time data streams from the cell phone carriers can replace the archived data sources used during development.

4.2 Detection, Simulation, and Prediction Layer

The *Detection and Alert System* (DAS) processes the data stream provided by RTDS, and after comparing it with the baseline data stored by HIS, discovers potential deviations from the normal communication and spatial location patterns. Anomalies are reported to the *Simulation and Prediction System* (SPS), which then accesses the detailed data available to RTDS, selects the relevant information in the vicinity of

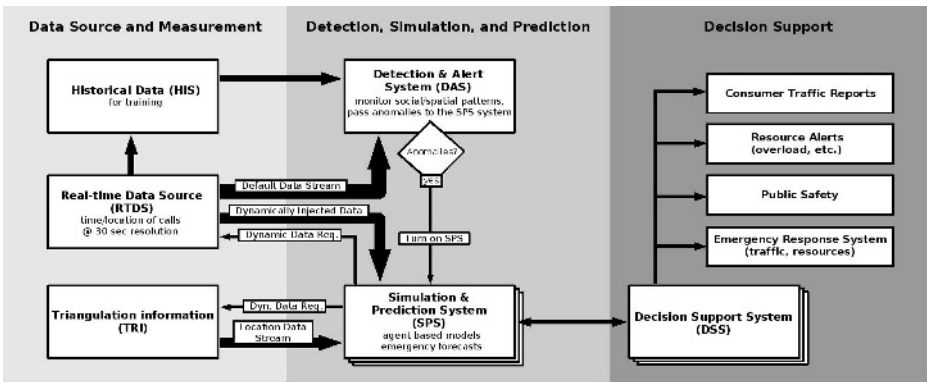


Fig. 1. The data-streams on the location and the communication patterns of the mobile phones are generated by RTDS, which passes the information to DAS, an application designed to discover potential traffic and communication anomalies by comparing the streaming activity data with a historical reference data stored in HIS. Anomalies are reported to SPS, which initiates a dynamic, data-driven agent-based simulation, predicting cell phone movement in the physical vicinity of the anomaly. SPS has the capability to request detailed time dependent coordinates of mobile phones in the vicinity of the anomaly from the TRI application. The predictions on the unfolding of the anomaly are sent in real time to DSS, which feeds it selectively into the relevant alert and monitoring services for use by emergency response managers.

each reported anomaly, and using a rule-based system, generates hypotheses about the nature of each anomaly. The SPS then launches one or more agent-based simulations for the reported anomaly; each simulation is a test of a hypothesis and able to predict the evolution of the anomaly. If the spatial resolution in the vicinity of the anomaly is not sufficient, SPS will instruct the Triangulation Application (TRI) to track the precise spatial location for each phone in the vicinity of the anomaly. The detailed location data from the RTDS and TRI will be used to dynamically test, update and validate the simulations. Multiple instances of the SPS may be invoked in response to multiple concurrent alerts.

4.3 Agent Based Simulations

Within the agent-based simulations of the SPS, each cell phone is represented as an agent. Each agent, based on the hypothesis the simulation is testing, may have unique properties and behaviors. For example, if the cell phones are in cars on a limited access highway, their freedom of movement will be limited to the road, freeway exits, and the other vehicles. Alternatively, if the cell phones are hypothesized to be carried by pedestrians walking through a city, perhaps fleeing a natural event (fire, explosion, etc.), their motion will be blocked by rivers, fenced-in highways, buildings, and other barriers to pedestrians. The agent-based simulation paradigm we use is a bottom-up approach, with large numbers of simple agents, interacting with each other and the simulated environment [4-7]. The implementation is objected-oriented, where each agent in the simulation is an instantiated object of a class or subclass of cell phone user-types, providing the software engineering benefits of inheritance, encapsulation and specialization. An agent-based simulation tool, RePast is used to build the simulations [8]. Repast is especially attractive since 1) it can be used in the Java/J2EE or .NET environments [9] providing maximum portability, and 2) has recently been extended to work with both the ESRI ArcMap [10] and OpenMap GIS (Geographical Information Systems) [11, 12]. This will permit the use of GIS map data that can 1) accurately represent mobile phone cells (i.e., the coverage of a cell tower) and 2) use actual topographical map data to constrain the movement of agents in the simulations. The simulations execute either by event scheduling or by time stepping the agents, per their rules of behavior and individual attributes. A history of each simulation is stored in a database, and as newer data is injected into the simulation system, each simulation's predictions up to that point in time will be dynamically validated against the new real data. Those simulations that are "close" to "reality" will be recalibrated and rerun from that point in time. Others that fail to correlate with the newly injected data will be terminated and new hypotheses and simulations generated and executed if needed. Validated simulations are then used to predict the evolution of the anomaly and this is shared with the modules in the DSS Layer.

4.4 The Decision Support System (DSS) Layer

The DSS is the user interface to the WIPER system, providing a view on the status and the predicted evolution of the emergency. It can activate response applications that the emergency response managers and autonomous systems may decide to invoke. These response systems might include information feeds to law enforcement,

public safety, customer data services, traffic reporting systems, emergency crews, Amber Alerts, and alert systems to the mobile phone company's own network engineers and managers. Emergency response managers can send requests to the SPS Layer for predictions and visualizations of the cell phone distributions in the area of the disaster. Similar to the SPS, multiple instances of the DSS may be invoked by different emergency response users of the WIPER system.

4.5 System Software Architecture

The tasks that WIPER will address – that of monitoring, detecting, analyzing, simulating, predicting, and responding to anomalies in the movement of large numbers of individuals, by tracking their cell phones' locations and call activities – will consist of multiple distributed heterogeneous applications that must be able to exchange data and control information, both in the real world deployment and in our prototype development, testing and evaluation. In the real world deployment some of these programs may execute within one or more cell phone companies, and others (DSS or response applications) may execute within government public safety (e.g., law enforcement, emergency response, and security) organizations. The data collection and reporting services must be able to respond to requests for more detailed data for cell phone tracking, and the monitoring and simulations must be able to adapt to new streaming data that may need to be injected into the running simulations. Decision support systems, either within the cell phone companies or within the public safety organization must also be able to receive situation data from the monitoring systems and predictions from the simulations and give users of the DSS the ability to compose workflows of services (data sources, simulations, predictions, visualizations, status reports) in an ad hoc and dynamic fashion. Likewise, in the proposed DDDAS development project, the HIS, RTDS, TRI, DAS, SPS, DSS and response systems will be developed and deployed using different tools, platforms, and servers. We are using the open standards-based Service Oriented Architecture (SOA) to integrate these systems and to enable the end-user composition of system modules from within the DSS. The SOA is being adopted both by industry and in e-government (e.g., cell phone service providers, public safety organizations, emergency response agencies) and the scientific and engineering communities (e.g., Globus, Grid Services, Service Oriented Science) [13-17]. The SOA employs open standards, such as the Open Grid Services Architecture (OSGA), extensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL) etc., from the Global Grid Forum (GGF), WWW Consortium (W3C), and OASIS standards bodies, and this will permit interested mobile phone companies, other interested researchers, and other industrial and government organizations to more easily reuse, integrate and extend our software. All programs, simulations, and online databases are: 1) written or wrapped as Web Services, 2) deployed to individual servers as needed for performance reasons and to emulate the real world distributed nature of this DDDAS application, and 3) communicate with one another in a loosely coupled distributed fashion by sending XML/SOAP formatted messages to one another as shown in Fig. 2.

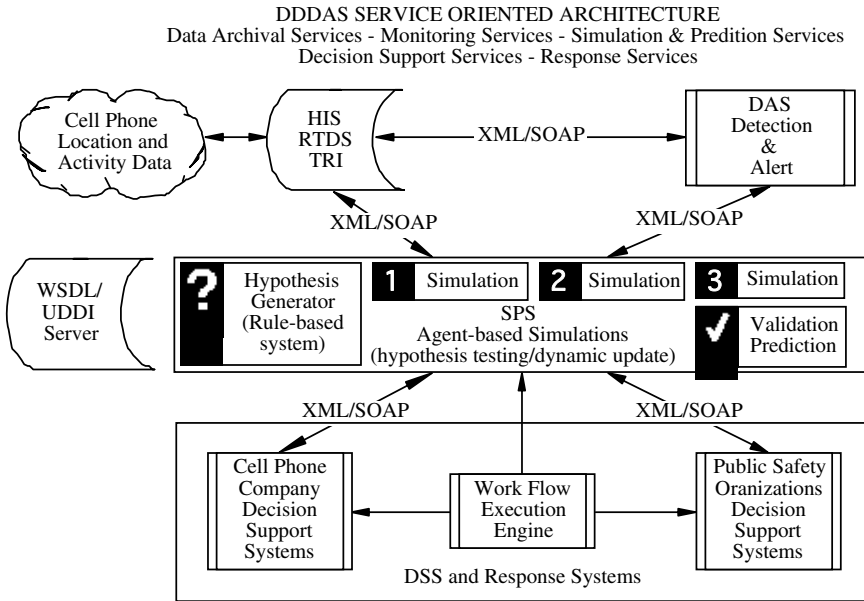


Fig. 2. The WIPER Service Oriented Architecture (SOA). The use of business and e-government standards based middleware is needed since the data suppliers are typically commercial enterprises and the users of the DSS will be typically governmental public safety organizations.

5 Discussion and Summary

Mobile phone companies routinely record the location and communication patterns of millions of cell phones. These cell phones form a large pre-existing *mobile wireless sensor net*, generating datasets of potential value to public safety managers, emergency response personnel, traffic engineers, city planning and resource management, offering a thorough snapshot of what humans do on a daily basis, how crowds self-organize, and how individuals alter their behavior when faced with emergencies, traffic jams, civil disorder or terrorist attacks.

WIPER is projected to be capable of real-time monitoring of normal social and geographical communication and activity patterns of millions of wireless phone users, recognizing unusual human agglomerations, potential emergencies and traffic jams. WIPER will select from these massive data streams high-resolution information in the physical vicinity of a communication or traffic anomaly, and dynamically inject it into an agent-based simulation system to classify and predict the unfolding of the emergency in real time. The agent-based simulation system will dynamically steer local data collection in the vicinity of the anomaly. Multiple distributed data collection, monitoring, analysis, simulation and decision support modules will be integrated using a Service Oriented Architecture (SOA) to generate traffic forecasts and emergency alerts for engineering, public safety and emergency response personnel.

Both the reliability of the cellular phone system during a disaster, and privacy concerns present potential limitations on the WIPER system. Since the WIPER system uses wireless cell phones to collect data about the population during an emergency, extreme large-scale disasters such as earthquakes and hurricanes can disable key components of the cell phone system, thus reducing its data collection ability in areas impacted most by the disaster. This was observed several days into hurricane Katrina in New Orleans (USA) when flooding disabled cell towers and the landline phone network they connect to [18]. At the individual level, since cell phones typically have battery standby times of a few days, prolonged power outages will reduce the number of phones available to collect data from.

Privacy concerns with government tracking cell phone locations and wiretapping present a challenge for the deployment of a system such as WIPER [19, 20]. Measures may be required to filter or anonymize the IDs of individual phone data prior to streaming it to the WIPER system to preserve individual privacy. The WIPER system does not use the individual identify of each cell phone, but only the aggregate numbers and calling patterns of the phone in the area of the emergency. As stated earlier in this paper, the WIPER development project is working with real but anonymized data, and has no access to the content of phone calls or messages. This project will continue to examine the impact on personal privacy that a system such as WIPER may have, especially in the context of GIS systems and technologies [21, 22].

References

References for this paper can be obtained by contacting the authors or online at:

<http://www.nd.edu/~dddas/Papers/papers.html#ICCS2006Refs>

Dynamic Data Driven Application Simulation of Surface Transportation Systems

R. Fujimoto, R. Guensler, M. Hunter, H.-K. Kim, J. Lee, J. Leonard II,
M. Palekar, K. Schwan, and B. Seshasayee

Georgia Institute of Technology, Atlanta, GA 30332, USA
{fujimoto@cc, randall.guensler@ce, michael.hunter@ce,
john.leonard@ce, schwan@cc}.gatech.edu

Abstract. A project concerned with applying Dynamic Data Driven Application Simulations (DDDAS) to monitor and manage surface transportation systems is described. Building upon activities such as the Vehicle-Infrastructure Integration initiative, a hierarchical DDDAS architecture is presented that includes coupled in-vehicle, roadside, and traffic management center simulations. The overall architecture is described as well as current work to implement and evaluate the effectiveness of this approach for a portion the Atlanta metropolitan area in the context of a hypothesized emergency evacuation scenario.

1 Introduction

The complexities associated with managing future transportation systems will predominantly reside in the interaction between sensing systems, data acquisition systems, demand models, communications, and traffic simulations. Systems soon to be launched under the Vehicle-Infrastructure Integration (VII) initiative will deploy a variety of roadside and mobile sensing platforms capable of collecting and transmitting transportation data [1-3]. Old paradigms associated with the use of basic information to identify incidents and mitigate recurrent congestion will fall by the wayside as new technologies come online. Centralized locations, local area hubs, and individual vehicles will receive fixed sensor and mobile vehicle data to collaboratively monitor and evaluate system performance, perform simulation runs, and plan routes through the system. Such systems will be interactive, iterative, and adaptive. As these systems evolve, they can be increasingly used in developing responses to rare or non-recurring conditions such as those occurring in emergency response scenarios.

As sensors proliferate, the amount of transportation-related data will increase by orders of magnitude. Yet, communications bandwidth and computer processing capabilities will be limited by availability and cost. Effective and efficient system management will require real-time determinations as to which data should be monitored, and at what resolutions. Distributed computing can help mitigate system limitations. Data collection, data processing, data analysis, and simulations performed by system agents (sub-network monitoring systems, base stations, vehicles, etc.) will lessen communication bandwidth requirements and harness surplus computing capacity. Middleware to manage the distributed network, synchronize data and results among autonomous agents, and resolve simulation output conflicts between agents

using disparate data sets become critical activities in such a system. With proper feedback, the connections between autonomous agent simulations will allow each simulation to adapt to better model reality.

2 A Motivating Example

The following scenario illustrates how the envisioned distributed traffic simulation system would operate in detecting and responding to an emergency. Suppose that a large explosion of unknown origin occurs. The vehicle-to-vehicle (V2V) sensor network normally used to transmit data for traffic operations management observes the loss of communication among multiple vehicles and roadside devices. Base stations on the arterial network (inside traffic controller cabinets) normally used to compile data from fixed detectors and instrumented vehicles and to simulate vehicle activity and traffic flow responses to changes in signal timing plans, identify the loss of communications among the devices within the local vehicular network. Instrumented vehicles, roadside sensors, and roadside data waypoints unaffected by the blast begin to record a significant disruption in traffic flow (decreases in speed, reductions in through volumes, etc.), reporting these data to local roadside nodes and to the regional traffic operations center (TOC). In response to the loss of sensors and controllers, the remaining upstream and downstream nodes automatically reconfigure to operate without the data from the affected vehicles and sensors, and without the simulation outputs of the affected nodes. Automated GIS routines at the TOC identify the location of the explosion based upon the data outage and transmit detailed site hazard information to the emergency management center (EMC).

The roadside base stations, in-vehicle computing systems, and other computers collectively process available network and agent data to establish an initial event boundary for use in emergency response planning. This facilitates the implementation of any pre-planned response initiatives (i.e. the initial deployment of emergency response personnel). The sensor network and distributed simulation system also allows for the implementation of a dynamic emergency response plan. In a dynamic response scenario, roadway base stations and in-vehicle systems (real-time V2V and fixed sensor data) identify immobile vehicles, recalculate roadway capacities, and simulate local traffic after the event. In-vehicle, local area and TOC based simulations are utilized to monitor traffic operations, forecast traffic flow, and derive an evacuation plan designed to clear vehicles from the impacted area, direct inbound traffic away from the impacted area, and provide inbound priority to emergency response vehicles. Navigation computers in vehicles outside the zone receiving data via the V2V and roadside network and are instructed to plan paths around the traffic impact zone, using data that are continually repackaged and distributed by the system to optimize zone clearance. Within minutes of the blast event, multiple sensor networks will have transformed monitoring data into useful information for emergency response planners and area evacuations.

We envision that systems will be composed of a heterogeneous collection of in-vehicle, roadside, and traditional computation and sensor nodes (e.g., TOC/EMC-based servers) that must analyze current system states, predict future states, and rapidly adapt to unexpected disruptive events on short time scales. These systems

will be constrained by available communications bandwidth, data transfer costs, and distributed computing processing capacity.

This project is developing adaptive techniques to meet the challenging QoS requirements of transportation systems, providing capabilities such as in-flight data processing, continual awareness of end user needs, and dynamic adaptation to platform changes. Modeling and simulation research is focused on developing real-time, agent-based, data driven simulation models to be deployed within the transportation infrastructure. These models are being designed so that they can be replicated and integrated with other models through a distributed simulation framework, as well as interact with centralized, larger-scale models running in traffic management centers on high performance computing platforms. Finally, we envision multiple layers of autonomous on-line simulations operating in vehicles, signal controllers, and traffic management centers designed for a variety of purposes and sharing state predictions and model results.

3 System Architecture

As sensor networks continue to proliferate communication systems will evolve to the point where the detailed traffic data from these sensor networks become ubiquitously available throughout the network. This data will facilitate real-time fixed location and in-vehicle simulation. A notional diagram of the envisioned system is depicted in Figure 1. The hierarchical architecture includes sensors, in-vehicle computer systems and other elements as end nodes. Intermediate nodes include computer systems housed in traffic signal controllers and other roadside equipment. Regional traffic management centers (TMCs) form the highest level of the hierarchy.

A major improvement in traffic monitoring is derived from the integration of data from onboard vehicle sensors. The high-resolution data provided from instrumented vehicles allows researchers to examine minute changes in real-time traffic flow and regional stability of traffic performance characteristics over time. Each vehicle will simulate its region of interest, downloading as needed data from roadside sensors, vehicle sensors, and traffic controllers that act as servers for local vehicular simulations. Each vehicle downloads the map of its regions of interest from the server as it travels along the roadway. It will cache this map for future use. It obtains information such as location, speed, and direction of travel of other vehicles in its surrounding regions through in-vehicle sensors, roadside servers, and potentially neighboring vehicles in real-time. Aggregate data such as vehicle density and flow rates from other regions of interest may also be obtained through the server. Based on this data, future projections are made using simulation. This projected data might be used to make routing decisions, for example. As new information is obtained, simulation parameters or state variables are updated and projected future states are modified accordingly. Results of the simulation are continuously uploaded to the server. Many vehicles may be simulating the same region and updating information to the server. The server aggregates the information obtained from roadside sensors as well as simulation results in order to provide an accurate picture of the region.

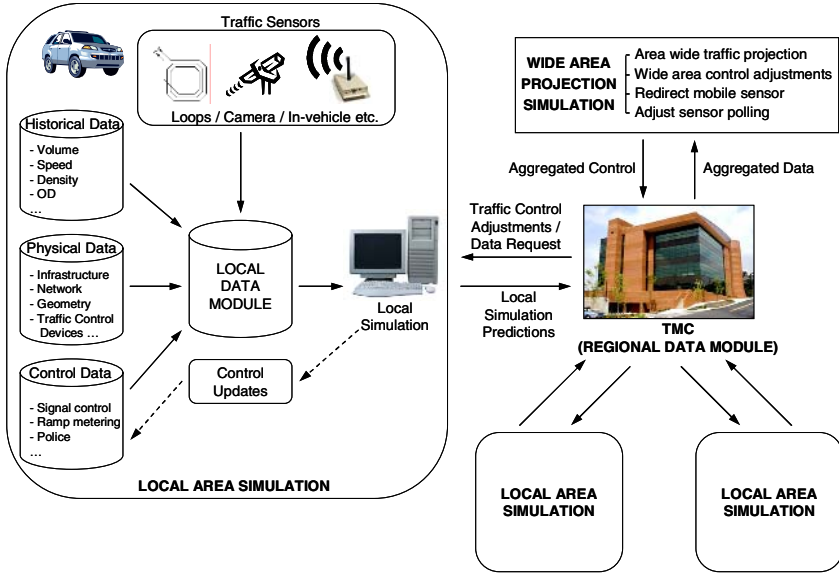


Fig. 1. Notional diagram of a dynamic, data driven intelligent transportation system

As communications-equipped vehicles continue to proliferate, their data will be extremely useful in monitoring real-time roadway operating conditions. A primary limitation encountered in integrating their data into the transportation monitoring network will be communications bandwidth. In addition, it is unlikely that centralized simulation and planning operations, such as those currently operated in regional traffic operations centers, will be able to integrate sufficient processing capacity to handle all of the potential data available within the transportation system, even if sufficient communications bandwidth were available. Thus, the types of systems being deployed as part of VII give rise to the need for a multi-level simulation architecture. Simulations will operate in at least three levels. In-vehicle simulations will typically be used for purposes such as route planning or travel time prediction. Traffic controller simulations will be used for purposes such as adjusting signal timings. The TMC simulation simulates larger regions, and can be used for city-wide operations planning and traffic monitoring.

We term systems such as these *ad hoc distributed simulations* because they are composed of loosely coupled, largely autonomous simulation, each modeling a different portion of the overall system, often at different resolutions. Unlike conventional distributed simulations that are created by partitioning the system into non-overlapping pieces (e.g., entities or agents), the simulators making up an ad hoc distributed simulation may model overlapping regions. Conflicts must be resolved. Further, if a simulation wishes to model new portions of the road network, it may obtain future state projections from other nearby simulators rather than recomputing this itself. A key research question that is being explored concerns the extent to which these ad hoc distributed simulations can collectively project accurate future states of the system.

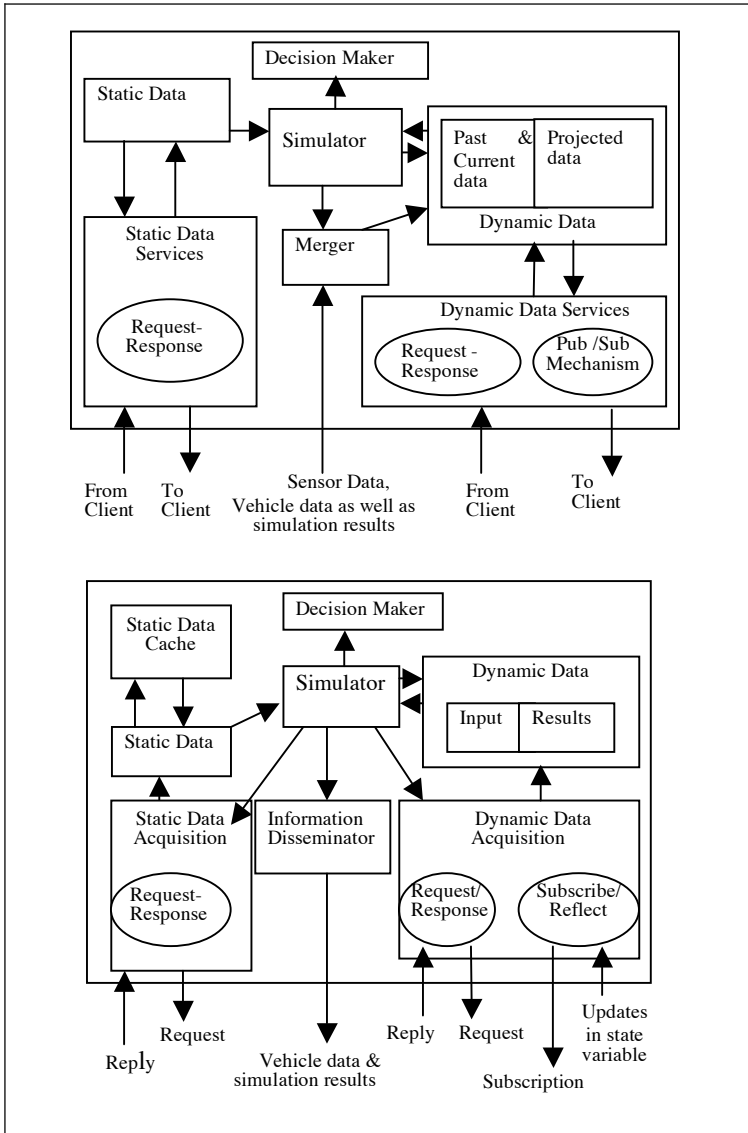


Fig. 2. Software components of (a) server (top) and (b) client (bottom)

The ad hoc distributed simulations considered here use a client-server architecture to coordinate amongst themselves. Traffic controller and TMC simulations act as servers in this framework, and propagate information up and down the hierarchy to obtain a global view of the entire system.

The communication problems arising out of mobility in a cooperative setting bear some similarity to those occurring in the robotics domain. Heterogeneous robot teams cooperate in mapping and exploring unfamiliar terrain [4], as well as in path planning

[5]. Due to the high failure of robots [6], as well as the need to conserve power, many adaptive algorithms have been proposed [7]. Such adaptive algorithms can find use in our efforts. Though the objectives are different, the middleware in both the domains has the common feature, in that it needs to adapt to a dynamic environment, and tolerate failure, to provide a reliable multi-node communication platform to the higher layers. Similar efforts are also being employed in the area of enterprise computing [8] where the middleware can detect dynamic behavior in the environment and automatically reconfigure to achieve the application's quality requirements.

A server (e.g., a simulator in a traffic signal controller or the TMC) in the mobile system architecture is depicted in Figure 2 (top). Static data includes information such as the road network that remains constant for long periods of time. Static data services are provided to clients through a request/response protocol. Dynamic data such as vehicle densities and flow rate continuously changes over time. Dynamic data can include both current and projected values obtained via simulations. Current values are obtained from sensors and clients. These values are used to calibrate the simulations and also predict future system states.

Projected data is used for decision making. This data is stored in the form of Time-Space Memory, i.e., values of state variables over different time values. Dynamic Data Services provide access to this data. Multiple simulations may simulate overlapping regions; the merger module is responsible for resolving inconsistencies and conflicts. The simulator is a server that predicts future system states. Results of the simulation are used for decision making. For example, if the server is present in the traffic controller then results may be used for signal time optimization.

The client architecture (Figure 2) includes similar components. Static and dynamic data is obtained from servers, and stored in a local cache for future use. Dynamic data can be broadly classified as input to the simulator and simulation results. Input data includes current values of state variables and is used to calibrate the simulation. Results of the simulation include the projected state. Vehicle data such as location, speed, route, etc., as well as simulation results are broadcasted by the vehicle to update the state variables at the server. The individual simulations incorporate live data into the simulations as updates occur. Model parameters must be changed depending on the real time data in order to accurately model the traffic system.

4 Transportation Simulation Models

Throughout the duration of an emergency situation high-resolution mobile, in-vehicle simulations may be utilized to help observe, control, and manage traffic operation in a local area. Two major distinctions are readily apparent between mobile, in-vehicle simulation and traditional transportation simulations. First, an in-vehicle simulation must be able to process a dynamically changing roadway topology. Throughout the duration of the simulation the roadway topology must be updated to represent the area focused around the vehicle. Current transportation simulation models assume a fixed roadway topology, the exception primarily being the modeling of road closures. Second, to be useful an in-vehicle simulation must have the ability to adapt to real-time inputs (volumes, signal indications, etc.), a feature uncommon in current transportation simulations. For example, an emergency scenario represents a non-recurring situation

with potentially dramatic changes in traffic patterns and traveler behavior. Facility operations (e.g. traffic signals, ramp meters), pedestrian movements, changing traveler destinations, police traffic control, etc. all will impact operations in a real-time manner. Even where potential emergencies are modeled well in advance it is highly unlikely that reliable predictive models will be obtained, as people may not follow the evacuation plan, pedestrians may not observe traffic controls, vehicles may not stay within lane lines, and off-road operation may occur.

In addition to the in-vehicle simulation models we are also utilizing existing microscopic simulation platforms to develop several models for combined and complementary experimental analysis. The first model is a CORSIM simulation that incorporates the northern half of the Atlanta area. (CORSIM – short for corridor simulation - is a microscopic, stochastic traffic simulation model developed by the Federal Highway Administration for modeling individual vehicle movements on a second-by-second basis for the purpose of assessing the traffic performance of highway and street systems.) The study area lies in three counties (Fulton, DeKalb, and Cobb) within the Atlanta metropolitan area. The freeway network within the study area includes 8 interchanges between freeways and 80 on/off ramps. The current model includes approximately 1,214 miles of surface streets and 289 miles of freeway.

Data requirements for a model of this size are significant, including network geometry (link length, free flow speed, grade, number of lanes, turn-bay et al.), traffic control (sign, signal control plan), and traffic flow data. The data for the CORSIM model was drawn primarily from the Atlanta Regional Commission's greater Atlanta travel demand model (number of lanes, link lengths, free flow speeds, the presence of an HOV lanes, turn movement ratios, and traffic flows, etc.), USGS one-foot resolution aerial photos (precise lane configurations, turn-bay presence and length, roadway curvature, on/off-ramp configurations, length of acceleration/deceleration lane, etc.) and state and local government agencies (primarily signal control and volume data). The current CORSIM network contains 3,227 nodes (1,940 surface nodes and 1,287 freeway nodes) and 5,974 links (4,764 surface links and 1,210 freeway links). A second model being developed covers a significantly smaller geographic area, centering on the Georgia Institute of Technology. This model is being developed at higher resolution, incorporating unsignalized intersections and major traffic origins/destinations. The Georgia Institute of Technology centered models will be utilized to support initial simulation and field experiments. Three separate simulation packages (CORSIM, VISSIM, and SimTraffic) are being coded, allowing for the exploration of potential bias introduced by selected simulations.

5 Project Status

At the time of this writing, implementations of three simulation models representing different levels of the simulation hierarchy (in-vehicle simulation, a regional Georgia Tech model, and a larger Atlanta area model) have been created. Further, runtime infrastructure middleware used for coupling the simulators has also been developed based on the client-server architecture described earlier, and an initial distributed simulation utilizing multiple in-vehicle simulations has been created. Middleware termed IFLOW offering data management functions such as processing of streaming

vehicle data and Quality of Service support has been developed. Other research is examining deployment of test systems in vehicles using mobile laptop computers and wireless communication infrastructure on the Georgia Tech campus.

References

1. Werner, J., Details of the VII Initiative 'Work in Progress' Provided at Public Meeting. 2005.
2. Bechler, M., W.J. Franz, and L. Wolf. Mobile Internet Access in FleetNet. in *KiVS 2003*. 2003.
3. Werner, J., USDOT Outlines the New VII Initiative at the 2004 TRB Annual Meeting. 2004.
4. Robert Grabowski, Luis E. Navarro-Serment, Christiaan J.J. Paredis, Pradeep K. Khosla, Heterogeneous Teams of Modular Robots for Mapping and Exploration, *Autonomous Robots*, Volume 8, Issue 3, Jun 2000, Pages 293 - 308
5. David Silver. Cooperative path-planning. In *AI Programming Wisdom*, 2006.
6. J. Carlson and R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, Volume 21, Issue 3, Jun 2005.
7. Keith J. O'Hara, Ripal Nathuji, Himanshu Raj, Karsten Schwan and Tucker Balch. AutoPower: Toward Energy-Aware Software Systems for Distributed Mobile Robots. *IEEE International Conference on Robotics and Automation*. 2006.
8. Karsten Schwan et al. Autonomic Information Flows. Technical Report GIT-CERCS-05-22. Georgia Institute of Technology. 2005.

DDDAS for Fire and Agent Evacuation Modeling of the Rhode Island Nightclub Fire

Alok Chaturvedi¹, Angela Mellema¹, Sergei Filatyev², and Jay Gore²

¹ Purdue Homeland Security Institute, Purdue University, West Lafayette, IN 47906
{alok, amellema}@purdue.edu

² School of Mechanical Engineering, Purdue University, West Lafayette, IN 47906
{sfilatyev, gore}@purdue.edu

Abstract. A Dynamic Data Driven Application System (DDDAS) was created to study interaction between fire and agent models during a fire evacuation. The analysis from that research can be used to validate proposed ideas in evacuation and building designs to ensure safety of buildings given various agent behaviors. Two separate models were used to simulate the components of the emergency situation: fire and agent. The independent models were able to run using data computed by the other interacting models, allowing careful examination of real-time interactions in a situation. Through study of the interactions, a better understanding is gained of how individual variables such as exit position and width affect the evacuation process and escape rate in the given scenario. Exits can be relocated and changed to quickly assess the effect on the model. The results can be used for improving building design and regulations as well as training first responders.

1 Introduction

In the event of an emergency situation, skilled planning and response preparation can dramatically affect the results of the scenario. In order to prepare a response for such an event it is important to study each important aspect of the model in order to understand how changes in the environment affect the outcome of the emergency situation. The scenario presented in this paper is modeled after The Station nightclub fire of West Warwick, Rhode Island which took place on February 20, 2003. In the case of a fire such as that described in the present paper the important aspects of the model are fire propagation and human evacuation. The fire and agent simulations are run simultaneously. By creating a virtual environment where simulations are run simultaneously the factors that affect each model can be monitored and measured to learn how to better prepare for and respond to an emergency situation.

Realistic models describing fire propagation and human behavior were needed for this comprehensive simulation. Currently, models exist which simulate fire and human behavior during evacuation, but those models have yet to be integrated to simulate real-time events and decisions [1]. As a result of studying key events of an emergency scenario, decisions can be made about emergency response and building planning prior to the event.

2 Models

The present scenario was modeled after The Station nightclub fire. The National Institute of Standards in Technology has completed a thorough investigation and computer simulation of that fire [2]. The research completed by NIST was used to validate the current project, and as a foundation for continued research. Figure 1 shows the basic floor plan of the nightclub.

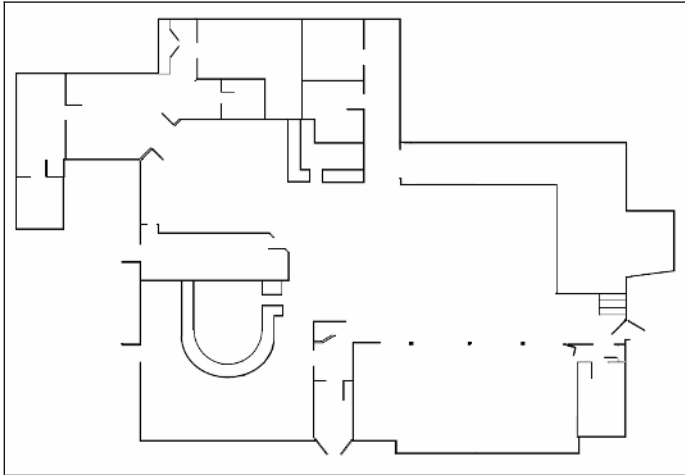


Fig. 1. Basic floor plan of The Station nightclub. The exit at the bottom of the picture is the main exit. The exit on the right was the exit closest to the stage.

There are four exits in the building: the main exit (bottom of Fig. 1), the stage exit (right of Fig. 1), the kitchen exit (top left of Fig. 1), and the bar exit (bottom left of Fig.1). The fire began on the stage, very close to the stage exit, and began as a result of a pyrotechnics display igniting polyurethane foam that was lining the walls of the stage and dance floor. The main exit was the exit used by the majority of the people inside of the building, due to a familiarity with that exit.

2.1 Fire

Fire was simulated using the Fire Dynamics Simulator, FDS4, developed by NIST [3]. FDS4 is a large-eddy simulation (LES) turbulent code which employs the mixture fraction model. Simulations run using FDS4 have been successfully used to describe different processes from fundamental combustion studies of plumes to the replication of the 2001 fire in the World Trade Center [4]. FDS provides time-resolved temperature, carbon monoxide, carbon dioxide and soot distribution throughout the building for the duration of the fire. These calculations show how the fire and smoke propagate through the building and the results were used in the agent evacuation model.

After the fire occurred at The Station nightclub, NIST used FDS to extensively simulate the progression of the fire [2]. Most of the materials that were present in the building at the time of the fire were tested and the combustive properties of those

materials were used within the FDS simulation. The input file used by NIST was also used for the fire simulation in the current paper [5].

2.2 Agent

The agent model was designed to simulate agent behaviors during evacuation. In the case of a fire in a densely populated nightclub, crowd effects were considered and implemented.

Agent movements within the environment were simulated using a generalized force model described by Helbing, Farkas & Vicsek [6]. Agents exist in an environment in which socio-psychological and physical forces are exerted upon each agent influencing their behavior. Each agent i is pulled toward their goal position, the exits, in a certain direction \mathbf{e}_i^0 at certain desired speeds v_i^0 according to their mass m , and this attraction determines their evacuation path. Each agent adapts their actual velocity v_i to a certain time τ_i . Interaction forces of other agents j and walls W are represented by \mathbf{f}_{ij} and \mathbf{f}_{iW} , and also control the velocity of each agent. The following equation was used to calculate acceleration.

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - v_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_W \mathbf{f}_{iW} \tag{1}$$

The forces exerted upon the agents from objects such as walls and agents also can affect the agent’s position. Two agents i and j tend to stay away from each other as they are moving in a given direction based on a repulsive force $A_i \exp[(r_{ij} - d_{ij})/B_i] \mathbf{n}_{ij}$, where A_i and B_i are constants.

$$\mathbf{f}_{ij} = \{A_i \exp[(r_{ij} - d_{ij})/B_i]\} \mathbf{n}_{ij} \tag{2}$$

Lastly, the agents interact with walls and other obstructions throughout the environment. d_{iW} represents the distance to the wall W , \mathbf{n}_{iW} represents the direction perpendicular to it. The final equation denoting the interaction between agents and other obstructions is given by:

$$\mathbf{f}_{iW} = \{A_i \exp[(r_i - d_{iW})/B_i]\} \mathbf{n}_{iW} \tag{3}$$

Agents choose to evacuate out of a particular exit based on the agent’s familiarity with the building. Different agents can be aware of certain exits based upon where they are located and any previous knowledge of the building. This is represented in the simulation by giving each agent a varying level of knowledge of the building’s exits. As the conditions in the building worsen the agent may chose to change their route based upon their personal safety. For example, as many agents are attempting to leave out of one exit the exit may become blocked and, as the conditions in the building worsen, agents may attempt to find another exit.

Work is currently being done to allow the fire to affect the health of each agent based on ISO standard 13571 [7]. Fractional Effective Doses (FED) for heat exposure

and gas concentration for each time t will be calculated and used to lower the health of the agents appropriately.

3 Model Interaction

Agents can take actions such as breaking windows and opening doors. These actions have an effect on how the fire moves throughout the model.

Currently agent's actions can only fuel fire by opening windows and doors, agents cannot take action against the fire. This functionality is in development. Since no significant fire fighting measures were taken in the current scenario, this implementation was not necessary.

4 Results

Three cases of the scenario were computed for the fire evacuation. The first two cases were similar to the cases used in the research conducted by NIST [2]. The first case was calculated allowing all agents to leave out of each exit and each agent chose the exit that was closest to them. The exit times of each agent were monitored, showing the best evacuation time in optimal building conditions, 226 seconds.

As the first case does not accurately represent the conditions in the building, the second case was designed to reflect the conditions of The Station nightclub more closely. In this case, the exit that is closest to the fire's point of origin is considered impassible at the time that the fire consumes the area, when time is $t=30$ seconds. The second exit that became impassible was the main exit, which became blocked due to crowd crush. This happened at $t=90$ seconds. When these two exits were blocked at their respective times, the time of evacuation grew to 346 seconds. Figure 2 shows the

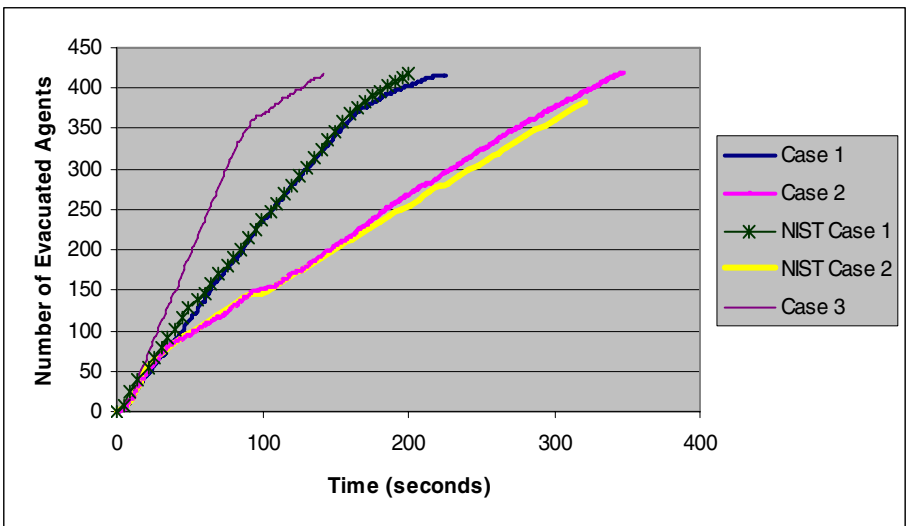


Fig. 2. The rates of evacuation for the first two cases of the scenario plotted with the rates of evacuation in the research completed by NIST

rates of evacuation during the two cases. The slope of the data line for Case 2 in figure 2 decreases at the time of the two exit closings, showing the affect of the closings on the overall evacuation rate.

The first two cases were replications of the NIST research, and were computed to benchmark the current system. These two cases corresponded very closely with the research completed by NIST. The blocking of the two exits in the second case affected the rate of evacuation similarly and the corresponding cases took roughly the same amount of time to evacuate.

A final case, Case 3 of Figure 1, was computed using the DDDAS to calculate the effect of modifying areas of the building that impede evacuation. Figure 3 shows how the main exit area was changed to calculate Case 3.

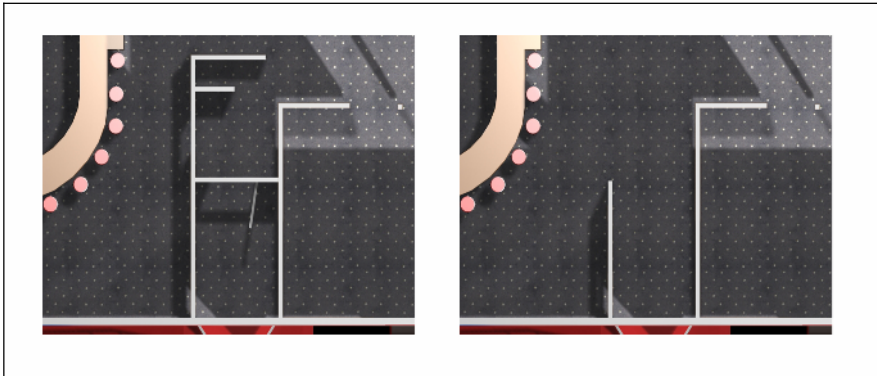


Fig. 3. The image on the left is the model used to represent the nightclub on the night of the fire. The image on the right is the modified model used to asses the affect on evacuation



Fig. 4. The image on the left is the evacuation visualization at 5 seconds; the image on the right is the evacuation visualization at 30 seconds. The Case 1 visualization is shown

This simple change to the geometry of the building greatly affected the amount of “crowd crush” that was experienced. The third case was a modification of the first case, in which the exits were not blocked at any time. Agents each chose the exit closest to them. In comparison to Case 1, all agents that were closest to the main exit were able to evacuate out of the main exit even faster than the agents were able to exit out of the stage exit. This is a considerable improvement over the first case. The total time of evacuation for the third case was 143 seconds, as opposed to the time of evacuation for the third case, 226 seconds.

The visualization of the models is an important part of the DDDAS. It allows us to see clearly effects of potential problems, e.g. exit blockage, and to help finding solutions to them. Figure 4 is the visualization at $t=5$ and $t=30$ seconds of Case 1.

The fire spreads rapidly throughout the building. The visualization shown in Figure 5 shows the rapid spread of fire throughout the main room of The Station nightclub.

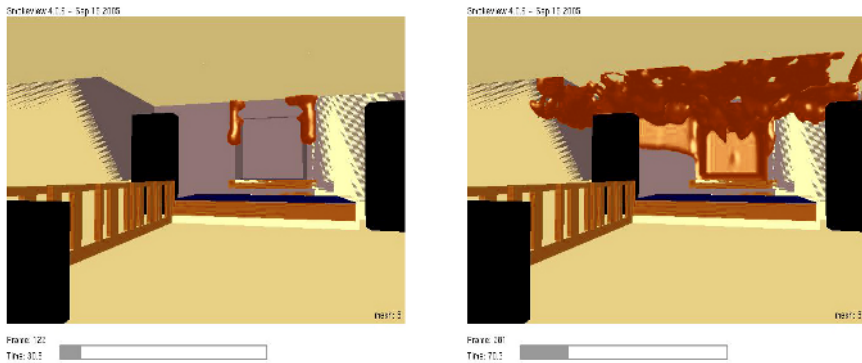


Fig. 5. Fire visualization. The image on the left is the visualization at 30 seconds; the image on the right is the visualization at 70 seconds.

Smoke is not shown in the above figure so that the flame position can be clearly seen. Just 70 seconds after ignition the stage area is consumed by flames and the main room is filled with smoke.

5 Conclusions

The Dynamic Data Driven Shared Reality System was created to study the events of The Station nightclub fire of 2003. Two scenarios were run to closely match the research completed by NIST shortly after the 2003 disaster. The results of those two scenarios gave similar results to draw the same conclusions as the NIST paper.

That research was then expanded upon to include the effect of fire upon the agents during evacuation. A third scenario was run testing how changes to the geometry can affect the total evacuation time as well as changes to the evacuation process. Simulations allow effects of these changes to be instantly visible and make it possible to set up many configurations of the building layout. Building planning, design and modification can then be considered more intelligently and also justified with data supporting

certain decisions. Real-time and near-real-time visualizations also assist with data collection and viewing the results of such changes.

References

1. Foong, C., Armstrong, B., Dilley, D., Grahn, J., Krull, K., Chaturvedi, A., Gore, J., Filatyev, S.: Towards Enabling a Distributed and Scalable Society of Simulations. 2005 Spring Simulation Multiconference (SpringSim '05). (2005)
2. Grosshandler, W., Bryner, N., Madrzykowski, D., Kuntz, K.: Report of the Technical Investigation of The Station Nightclub Fire. NIST NCSTAR 2, Vol. 1 (2005)
3. McGrattan, K., Forney, G.: Fire Dynamics Simulator (Version 4). Users Guide. NIST Special Publication 1019. 2004 <<http://fire.nist.gov/fds>>
4. Prasad, K., Baum, H.: Coupled Fire Dynamics and Thermal Response of Complex Building Structures. *Proc. Combust. Inst.* 30 (2005) 2255-2262
5. Grosshandler, W., Bryner, N., Madrzykowski, D., Kuntz, K.: Report of the Technical Investigation of The Station Nightclub Fire: Appendices. NIST NCSTAR 2, Vol. 2 (2005)
6. Helbing, D., Farkas, I., Vicsek, T.: Simulating Dynamical Features of Escape Panic. *Nature*, Vol. 407. (2000) 487-490.
7. International Organization for Standardization: Life-threatening components of fire – Guidelines for the estimation of time available for escape using fire data, ISO/TS 13571:2002. (2002)
8. Purser, D.A.: Toxicity Assessment of Combustion Products, SFPE Handbook of Fire Protection Engineering, P.J. DiNenno, Ed., 2nd Ed., National Fire Protection Association, Quincy, MA, Sec.2, (1995) 85-146

Auto-steered Information-Decision Processes for Electric System Asset Management

James D. McCalley, Vasant G. Honavar, Sarah M. Ryan, William Q. Meeker,
Ronald A. Roberts, Daji Qiao, and Yuan Li

Iowa State University, Ames, IA 50011, US
{jdm, honavar, smryan, wqmeeker, rroberts, daji,
tua}@iastate.edu

Abstract. The total replacement value of the US transmission lines alone (excluding land) is conservatively estimated at over \$100 billion dollars [1] and triples when including transformers and circuit breakers. Investment in new transmission equipment has significantly declined over the past 15 years. Some of the equipment is well beyond intended life, yet is operated under increasing stress, as load growth, new generation, and economically motivated transmission flows push equipment beyond nameplate limits. Maintaining acceptable electric transmission system reliability and delivering electric energy at low energy prices requires innovations in sensing, diagnostics, communications, data management, processing, algorithms, risk assessment, decision-making (for operations, maintenance, and planning), and process coordination. This paper overviews a comprehensive approach to develop methods and processes in these areas, driven by the ultimate objective to develop a hardware-software prototype capable of auto-steering the information-decision cycles inherent to managing operations, maintenance, and planning of the high-voltage electric power transmission systems.

1 Introduction

In electric power transmission systems, the assets include transmission lines, support structures, transformers, power plants, and protection equipment. Condition information includes loading or operating histories, inspection data, periodic and as-needed testing and diagnostic results, and continuous diagnostic measurements, the latter of which are typically collected via intelligent electronic devices (IED) and stored within substation servers. A single transmission company, each of which has their own centralized control center, has responsibility for many thousands of each equipment type. A single control area, represented by an Independent System Operator (ISO), oversees and coordinates activities of a number of different transmission companies. The eastern and western US interconnections are each comprised of a number of ISOs; the only other US interconnection, Texas, has only one. Failure of an asset may affect physical and economic performance of the entire interconnection and always increases likelihood of additional failures. Because economic performance (power supply allocation among power plants) affects transmission loading which affects failure likelihood and consequence, operational

risk-reduction inevitably results in less economic power supply. Frequency and severity of blackout scenarios as observed on August 14, 2003 are affected by policies associated with equipment operation, maintenance, and planning.

The objective in this work is to develop a hardware-software prototype capable of auto-steering the information-decision cycles inherent to managing operations, maintenance, and planning of the high-voltage electric power transmission systems. We focus on the needs of the most critical electric transmission equipment, including power transformers, circuit breakers, and transmission lines. Similar equipment exists at the distribution level, so the work will find direct application there. Figure 1 illustrates the structure of the problem and facilitates description of how we intend to approach its solution. We overview intended implementation of the 5 different layers in what follows.

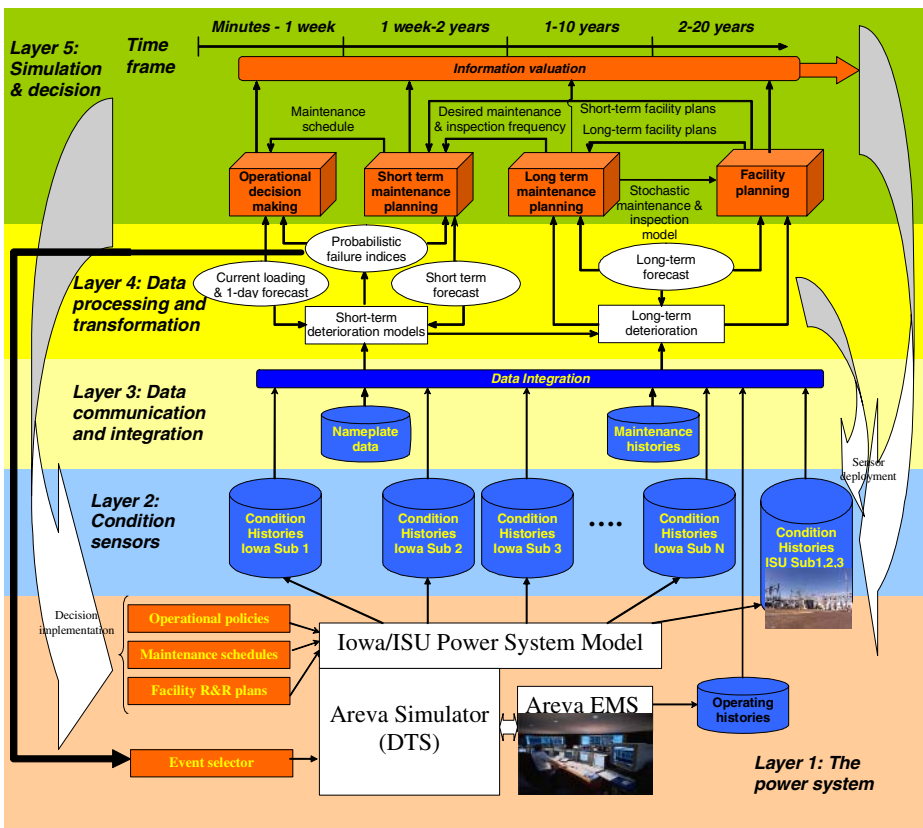


Fig. 1. The structure of the asset management problem

Layer 1, The power system: The prototype will center on a continuously running model of the Iowa power system using network data provided by local utility companies using a commercial-grade (Areva) simulator.

Layer 2, Condition sensors: As indicated by the taller “Condition History” cylinder at the far right of layer, 3 campus substations will be equipped with sensors, communication equipment, and servers to provide a benchmark prototype for hardware implementation. Other substations will be represented virtually, each with its own unique database containing the condition data for that substation provided by the utility company.

Layer 3, Data communication and integration: This will entail intra-substation communication using wireless between IEDs and substation server together with federated data integration to provide efficient, dependable, and secure mechanisms for interfacing Layer 4 data transformation algorithms with the data resources.

Layer 4, Data processing and transformation: This layer will operate on the integrated data from layer 3 to produce, for each component/failure mode/time, an estimate of that particular component/failure mode deterioration level at the given time. This will require deterioration models, and we target such models for the chemical degradation processes in oil and cellulose (both of which provide insulation in power transformers). We will also need stochastic models to predict future degradation, and we further describe these models in Section 2.

Layer 5, Simulation and decision: This layer will utilize the component probabilistic failure indices from layer 4 together with short and long-term system forecasts to drive integrated stochastic simulation and decision models. These models will operate interactively, so that simulation and decision in each time frame utilizes information from simulation and decision within other time frames. Resulting operational policies, maintenance schedules, and facility reinforcement plans will then be implemented on the power system (as represented by the Areva simulator). The decision models will also be used to discover the value of additional information. This valuation will be used to drive the deployment of new sensors and redeployment of existing sensors, impacting Layer 2. This layer is further described in Section 3.

2 Layer 4: Data Processing and Transformation

Component condition, deterioration level, or propensity to fail, is essential information for asset management decision problems. Our objective in this part of the work is to develop methods of computing component (or subsystem) failure probabilities. One unique aspect of this work is that in addition to steady-state failure probabilities that capture average behavior over a large number of components and over an extended period of time, we also require transient failure probabilities to capture instantaneous behavior for each specific component.

Consider a set of condition vectors $c(t)=[c_1(t),c_2(t),\dots,c_K(t)]$ for K similar components taken over an extended period of time $t=0,1,\dots,T$, where each vector $c_k(t)$ provides M different measurements $c_{k1}(t), c_{k2}(t),\dots,c_{kM}(t)$, on component k characterizing its condition at time t . The total possible number of measurements is less than $K\times T\times M$ because there are different frequencies for which different measurements are taken. We will augment $c(t)$ with operational and environmental information in building predictive failure models. For some system components, failure is closely related to a single condition measurement that can be measured over time and modeled in a manner that allows reasonably accurate prediction of failure

(e.g., extent of vegetation growth or the amount of chemical degradation). Let $c(t,e;\beta)$ denote the expected level of degradation for a unit subjected to environmental conditions e , where β is a vector of unknown model parameters to be estimated from available data. The form of the function c may be suggested by physical-chemical theory, (see, for example, [2,3,4]), past experience, or the available data. A failure-time cdf $F(t)$ is induced by a specified model for $c(t,e;\beta)$, the environment e , and a definition of failure (usually a specified value c_f , beyond which failure is said to have occurred). Stochastic behavior in $c(t,e;\beta)$ can be captured either by using a stochastic process model (e.g., [5]) or by driving a deterministic model with a stochastic environmental model (e.g., [6]). As new condition information is received for a given unit, it is possible to update the failure probability for that unit. For the special case in which all units are in a common and constant environment, [7] develops a model to describe the effect that nondestructive inspections will have on the failure probability. It is possible to generalize this “degradation analysis” approach to a vector of condition measurements, but statistical modeling of the joint distribution of a vector of condition measurements is more difficult, especially if the dimension exceeds 2. The Markov modeling approaches discussed next provide a useful alternative.

We can often characterize boundary conditions separating J states of deterioration in component k in terms of the measurements $c_k(t)$, via a deterioration function $g[c_k(t)]$. The deterioration function returns a deterioration level j identified by $d_{j-1} < g[c_k(t)] < d_j$, where the last state $j=J$ represents the failed state. State J need not represent the rare “blue smoke” condition where the component has catastrophically failed (and for which little data is typically available). Rather, state J represents a set of measurement values for which engineering judgment indicates the component should be removed from service. This approach to computing failure probabilities is illustrated in Fig. 2, based on multistate Markov models, where each of J states is represented as a deterioration level. The representation of Fig. 2 shows $J=4$ deterioration levels, and deterioration level j is reached only from deterioration level $j-1$. Yet, the model is flexible; any number of deterioration levels can be represented, and, if data indicates transitions occur between non-consecutive states (e.g., 1 to 3), the model can accommodate this.

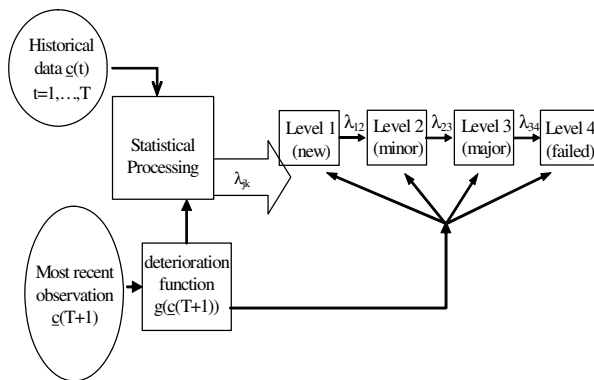


Fig. 2. Computing contingency probability reductions

The model parameters capture the deterioration in equipment state as influenced by past loading and environmental conditions. To capture future effects of variation in such conditions on model parameters, one needs to model the dependency of the transition intensities on these parameters. To account for uncertainty in state identification through observation or indirect measurement, the variance of the conditional probability of observation given the state, referred to as σ_{ki} for component k , state i is used. This parameter can also facilitate the analysis to identify investments to make for obtaining better or more information, described in Section 3.2.

Once transition intensities are determined, state probabilities are obtained from the transition probability matrix and initial state vector. We denote this failure probability for the k^{th} component as $p_k(c)$, a function of the time-dependent physical condition of the equipment $c(t)$. This modeling provides the ability to predict the effect that maintenance will have on failure probability and expected time to failure, metrics that are important for a number of decision problems. The expected time to failure is captured by computing first passage times [8,9].

3 Layer 5: Simulation, Decision and Information Valuation

Asset management decision problems are characterized by: (1) strong interdependencies between physical performance of individual assets, physical performance of the overall system, and economic system performance; (2) limited resources; (3) important uncertainties in individual component performance, system loading conditions, and available resources; (4) multiple objectives. We describe these decision problems in this section, together with our intent to solve them in an integrated fashion.

3.1 Simulation and Decision

Asset management decision problems can be classified into one of 4 types which all involve resource allocation with the objective to minimize cost and risk. These specific asset management decision problems include (a) Operations, (b) Short-term maintenance selection and scheduling, (c) Long-term maintenance planning, and (d) Facility planning.

These problems differ primarily in their time scale but are linked by a common focus on the interactions between the condition of equipment and the decisions taken. The operational decision problem of how to meet demand in the next hour to week treats facilities available and their deterioration levels as given (though the deterioration is not known precisely). The contribution here is to use condition measurements to more accurately estimate short-term failure probabilities along with the deterioration effects of loading each piece of equipment at various levels, and to integrate these improved estimates into the dispatch and unit commitment decisions. The tactical decision problem to allocate resources for maintenance in the next 6-24 months suppresses detail about hourly operations but considers an aggregate description of equipment loading when deciding how to allocate resources to best manage the condition of the equipment. Our approach will use historical data to better judge the combined effects of maintenance and loading on the equipment

deterioration and use this information to improve maintenance scheduling. The long-term maintenance problem examines tradeoff between maintenance expense and equipment life to find inspection and maintenance policy to minimize expected long run cost of keeping the equipment in service reliably. The strategic decision problem to both expand capacity and replace equipment over the next 2-20 years takes as input distributions of equipment life lengths resulting from adopted maintenance policies and determines when to replace existing equipment and invest in additional assets. Unlike previous models for equipment replacement and capacity expansion, we consider the cumulative effect of power flow on equipment life and take advantage of better data-driven life length predictions.

When we integrate these optimization problems having different time-scales together, we will treat the quantities that vary much more slowly as static and model quantities that vary much faster in a way that ignores the details of their variations, such as by replacing fast-moving quantities by their averages. A similar strategy is used in hierarchical planning of manufacturing systems [10].

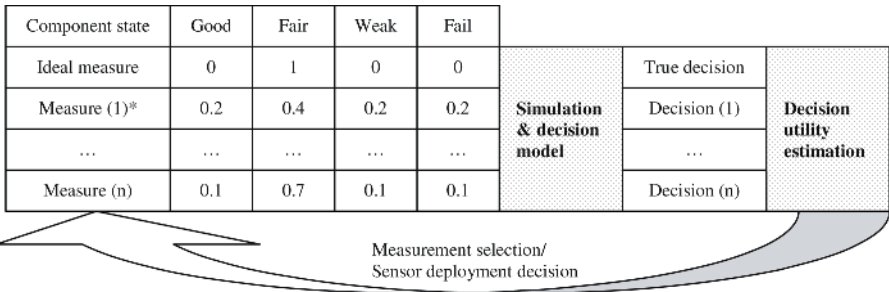
3.2 Information Valuation and Sensor Deployment

A last but not critical decision problem to be addressed is how to extract from data transformation (Layer 4) and decision (Layer 5) algorithms identification of economically sound opportunities for obtaining better information, thereby reducing uncertainty and improving decision-making capability. A simple case is when an abrupt measurement change causes immediate suspicion that a failure is imminent. The response is to inspect the equipment. Additional more specialized measurements may be done, and if those measurements confirm a problem, the equipment is removed. Such situations are addressed via alarms. A decision to obtain information is clear in this case, because imminent failure poses high capital loss and physical harm to humans.

Decision to gather information is more difficult for maintenance and planning problems because the payoff (or avoided loss) is not so pronounced. We address this problem via a two-stage information valuation approach [11,12,13,14]. In the first stage, we determine candidate components for which additional information may be of interest. Denoting the value of the objective function at the solution as Γ , we compute an index giving sensitivity in Γ due to component k , as $\sigma_{ki}(\partial\Gamma/\partial p_k)$ where p_k is the failure probability of component k and σ_{ki} (see end of Section 2) is the deviation in the observation for component k given that it is in state i . Components having high index are candidates to consider in the information valuation stage. Other selection criteria can be considered, e.g., we could identify components that are almost or barely selected by the decision algorithm. We denote additional information associated with candidate component k as r_{km} , which indicates component k is in state m . Such information may be obtained by installing more or better sensors at a cost. For example, a 50 year-old transformer that is a clear candidate for replacement may be operating with no monitoring equipment, yet installation of such equipment, providing information r_{km} , may result in decision to operate the unit for more years.

Following [13], states in which component k may reside are identified by $i=\{1,\dots,S\}$, with each state having probability π_i obtained from procedures described in Section 2. Simulation and decision algorithms are then repeated once for each

possible state of component k , generating solutions with corresponding objective values. Denote identified solutions (alternatives) by $a=\{1,\dots,A\}$. Thus, for each combination of state and alternative we have a consequence $c(a,i)$. Denoting the utility of an alternative as $u(a)$ and of a consequence as $v(c)$, we desire to choose the alternative to maximize expected utility $u(a,\pi)=\sum_{i=1,S}[\pi_i v(c(a,i))]$. The decision to obtain additional information is based on expected utility gains from shifting to better choices among the set of actions. Denote a_0 as the optimal alternative with no additional information, identified using prior probabilities π_i , and a_m as the optimal alternative with the additional information r_{km} . Then the value of information r_{km} is given by $\Delta(r_{km})=u(a_m, \pi_{i,m})-u(a_0, \pi_{i,m})=\sum_{i=1,S}[\pi_{i,m} v(c(a_0,i))]-\sum_{i=1,S}[\pi_{i,m} v(c(a_m,i))]$, where the posterior probabilities $\pi_{i,m}$ are given by $\pi_{i,m}=\Pr\{r_{km}|i\}\pi_i/\Pr\{r_{km}\}$. However, the decision to seek the additional information must be done *ex-ante* to be useful, and so



* Current measurement.

Fig. 3. Illustration of information valuation and sensor deployment

we cannot know that we will obtain r_{km} , i.e., that we will learn that component k is in state m . But we can assess (subjectively, or from historical data) the probability of learning from the new information that the component is in state m , which is $\Pr\{r_{km}\}$. Then we may compute the expectation of the value associated with the new information as $E\{\Delta(r_{km})\}=\sum_{m=1,S}\Pr\{r_{km}\}[u(a_m,\pi_{i,m})-u(a_0,\pi_{i,m})]$. We will use this approach to interface with Layers 4 and 5 procedures for assessing where and when to obtain additional information.

4 Conclusions

This paper gives a framework of a hardware-software prototype capable of auto-steering the information-decision cycles inherent to managing operations, maintenance, and planning of the high-voltage electric power transmission systems. The framework is divided into 5 layers and described in this paper accordingly. Although each layer represents an essential and substantive part of the framework, the paper focuses on the data transformation (in layer 4) and decision (in layer 5) elements.

Acknowledgments

The work described in this paper is funded by the National Science Foundation under grant NSF CNS0540293.

References

1. M. Gallaher, S. Johnston, B. Kirby, "Changing measurement and standards needs in a deregulated electric utility industry," Nov., 1999, Research Triangle Institute, <http://www.rti.org/pubs/dereg.pdf>.
2. Lu, C.J., and Meeker, W.Q. (1993), Using Degradation Measures to Estimate a Time-to-Failure Distribution. *Technometrics* 35, 161-174.
3. Meeker, W.Q., and LuValle, M.J. (1995), An Accelerated Life Test Model Based on Reliability Kinetics. *Technometrics* 37, 133-146.
4. Meeker, W.Q., Escobar, L.A., Lu, C.J. (1998), Accelerated Degradation Tests: Modeling and Analysis. *Technometrics* 40, 89-99.
5. Grall, A., Dieulle, L., Berenguer, C., and Roussignol, M. (2002), Continuous-Time Predictive-Maintenance Scheduling for a Deteriorating System, *IEEE Transactions on Reliability* 51, 141-150.
6. Meeker, W. Q., Escobar, L. A., and Chan, V. (2002), Using Accelerated Tests to Predict Service Life in Highly-VARIABLE Environments. Chapter 19 in *Service Life Prediction Methodology and Metrologies* Washington: American Chemical Society, J. W. Martin and D. R. Bauer, Editors.
7. Garrigoux, C.G. and Meeker, W.Q. (1995), Assessing the Effect of In-Service Inspections on the Reliability of Degrading Components, *Recent Advances in Life-Testing and Reliability*, N. Balakrishnan (Editor), CRC Press.
8. G. Anders, *Probability Concepts in Electric Power Systems*, John Wiley, New York, 1990.
9. A. Leite da Silva and J. Endrenyi, "Application of First Passage Times in the Markov Representation of Electric Power Systems," *Proceedings of the 4th International Conference on Probabilistic Methods Applied to Power Systems*, Rio de Janeiro, Brazil, Sept. 1994.
10. S. B. Gershwin, "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems," *Proceedings of the IEEE* Volume 77, Issue 1, Jan. 1989 Page(s):195-209.
11. H. Raiffa, "Decision analysis: Introductory lectures on choices under uncertainty," Addison-Wesley, 1968.
12. J. Mnarschak, "Economic information, decision, and prediction," Volume II, D. Reidel Publishing Co., 1974.
13. A. Baker, "Business decision-making," Croom-Helm, 1981.
14. J. Hirshleifer, "Time, Uncertainty, and Information," Basil Blackwell, 1989.

Data-Driven Power System Operations^{*}

E.H. Abed¹, N.S. Namachchivaya², T.J. Overbye², M.A. Pai²,
P.W. Sauer², and A. Sussman¹

¹ University of Maryland, College Park MD 20742, USA

² University of Illinois, Urbana, IL, 61801, USA

Abstract. In operations, simulation and control of power systems, the presence of real-time data relating to system states can yield precise forecasts and can enable robust active control. In this research we are developing efficient and robust methods to produce “data enhanced” reduced order models and filters for large-scale power systems. The application that this paper focuses on is the creation of new data-driven tools for electric power system operation and control. The applications systems include traditional **SCADA** systems as well as emerging **PMU** data concentrators. A central challenge is to provide near real-time condition assessment for “extreme events,” as well as long-term assessment of the deterioration of the electrical power grid. In order to provide effective guidance for power system control, we are also developing visualization methods for integrating multiple data sets. These visualization methods provide an up-to-date view of the system state, and guide operator-initiated power system control.

1 Introduction

The current power system operation paradigm relies heavily on two components: human operator experience and model-based analysis. Sensors currently exist at hundreds of thousands of locations dispersed at generating stations, at major network substations, and at loads. This sensor data is currently transmitted by Intelligent Electronic Devices (**IEDs**) to local and area control centers through Remote Terminal Units (**RTUs**) by a two-way network called a Supervisory Control and Data Acquisition (**SCADA**) system. In the past few years the restructuring of the electric power industry has created Regional Transmission Organizations (**RTOs**) and Independent System Operators (**ISOs**) that may also have access to certain amounts of this data. The area control centers, **RTOs**, and **ISOs** currently utilize this data in raw form for status displays and alarming. This data is also used as the input to a system State Estimator (**SE**) that is designed to filter erroneous data, create topology estimates, and provide the *best estimate* of the current state of the network configuration and the values of key indicators (voltages, currents, power and frequency). Human operators make judgments and decisions based on this data and on traditional analysis tools that utilize this data. At the very local level, automatic protection systems

^{*} This research is supported by NSF Grant #CNS-0540216.

react to measured data as they are programmed, with their primary objectives being to save equipment from damage in the event of abnormal conditions (lightning strikes, short circuits, etc.) and to continue to serve customers even under component failures.

The main objective of this research is to develop new algorithms and tools for the distributed collection, sharing, and harnessing of data for cooperative health monitoring and security assessment of power systems *in real time*. Fundamental concepts and technology for achieving this objective exist today in the areas of power systems, system theory and computer science, but harnessing them to achieve this vision requires the significant multidisciplinary concerted effort presented here. This work aims at making fundamental contributions to the necessarily multidisciplinary systems required for data utilization in monitoring and control of interconnected engineering systems. The application we focus on is the creation of new data-driven tools for electric power system operation and control. The systems include traditional **SCADA** systems as well as emerging Phasor Measurement Unit (**PMU**) data concentrators. Traditional **SCADA** equipment provides massive amounts of data that can be processed for diagnoses, awareness indicators, and other new operational tools. This effort will focus on data-driven tools for power system operations, which includes filtering, low-order modeling, and automated visualization for power system health monitoring and security assessment.

2 Data-Driven Tools for Power System Operations

Power system state estimation, security analysis, and real-time operation are currently heavily dependent on model parameters and assumed topology information. With the restructuring of power grid operations, the large geographical nature of the interconnected system is making the computation of state estimation more challenging. In addition, the massive volume of data and real-time alarms is making situational awareness and decision-making issues more severe. At the same time, new technologies are emerging to provide new types of data that can be utilized for both monitoring and control.

2.1 Signal Process: Power System Model

As the Data Grid maps the Power Grid in real time, it would be useful to have a view into the future evolution of the various variables. The traditional, time tested way to simulate the dynamic behavior of a power system is from a mathematical parameter-driven model of the system. The difference between the actual measured data and the simulated data can be used to improve the accuracy of the model. A typical dynamic model is an event-driven system so that we have a large-scale hybrid differential algebraic equations(**DAE**) system of the form:

$$\dot{X}_t = f(X_t, Z_t, \lambda), \quad X_0 = x \in \mathbb{R}^n, \quad Z_0 = z \in \mathbb{R}^m \quad (1)$$

$$0 = \begin{cases} g^1(X_t, Z_t, \lambda), & s(X_t, Z_t, \lambda) < 0, \\ g^2(X_t, Z_t, \lambda), & s(X_t, Z_t, \lambda) > 0. \end{cases}$$

In equations (1) a switching occurs when the switching function $s(x, z, u) = 0$. For example, if a device is switched into service at time $t = t_{sw}$, then the switching function in this case can simply be defined as $s(x, z, u) = t - t_{sw}$. In the above model, x are the dynamic state variables such as machine angles, velocities, etc.; z are the algebraic variables such as load bus voltage magnitudes and angles; and λ are the system parameters such as line reactances, generator mechanical input power, and fault clearing time. Note that the state variables x are continuous while the algebraic variables can undergo step changes at switching instants. The initial conditions for equation (1) are such that z satisfies the equation

$$g(x, z, u) = 0. \quad (2)$$

The model in equation (1) is imperfect. We assume that system (1) is subject to small *random jerks*, a fast perturbation process ξ that represents the unmodeled dynamics of the system, which we assume to be a Markov process. Then equations (1) become

$$\dot{X}_t = f(X_t, Z_t, \xi_t; \lambda), \quad X_0 = x \in \mathbb{R}^n, \quad (3)$$

$$0 = g(X_t, Z_t; \lambda), \quad Z_0 = z \in \mathbb{R}^m.$$

2.2 Observation Process: Phasor Measurement Unit

The introduction of **PMU** technology on a wide-area basis is offering a fruitful direction for dynamic data utilization. The **PMU** technology is providing Global Positioning System (GPS) based time stamping of key power system states, including phase angles of bus voltages. The **PMU** observation process $\{Y_t : t \geq 0\}$ is a function of (X_t, Z_t) corrupted by noise η_t , and is given by

$$Y_t = h(X_t, Z_t) + \eta_t \quad (4)$$

where h is called the sensor function that transforms the model variables into observed variables. Assuming data gathering and concentration at the rate of 60 measurements per second, this information can be used to detect dynamic trends as well as steady-state conditions. Initial research in this area indicates that the dynamic filtering of the **PMU** is very nearly equivalent to the mathematical filtering of the singular perturbation process in traditional power system dynamic modeling. This is important because it creates a one-to-one link between the dynamic model states and the measured data. This makes the data a valuable input for model validation, post-mortem analysis, and remedial action schemes in real-time control. Since most dynamic modes of interest are considerably slower than 60 Hz, these measurements will be useful in predicting damping of critical modes of oscillation.

2.3 Filtering

The theory of nonlinear filtering forms the framework in which problems of data assimilation for the nonlinear power network models will be treated. In this work, we consider the situation where the nature of the monitored environment will be captured by a Markovian state-space model that involves potentially nonlinear dynamics, nonlinear observations, and Gaussian observation noises. Our goal is to perform sequential estimation of the current system state from equation (2) at multiple sensor nodes from equation (4) in the power network system. In a nonlinear environment, particle methods, which includes sequential Monte Carlo and interacting particle filters are very useful. We will design distributed particle filters (or sequential algorithms) that can dynamically fuse the information recorded by the sensors without requiring excessive exchange of data. This work is being initiated through deterministic analysis of data measurements that determine initial conditions for model-based dynamic simulations. The results will be used in security assessment algorithms.

Using the algebraic equations (2), the variable Z_t can be eliminated in the model (1), as well as in the observer (4). Hence, in a discrete state space version of the model the state $x_k \in \mathbb{R}^n$ and the observation $y_k \in \mathbb{R}^p$ is described by

$$x_{k+1} = f_k(x_k, \xi_k) \quad (5)$$

$$y_k = h_k(x_k, \eta_k), \quad (6)$$

where $f_k : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $h_k : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$ are possibly nonlinear functions of the state x_k . The process and observation errors are represented by Gaussian white noise sequences w_k and v_k respectively and assumed to be independent to each other. Given the initial density $p(x_0)$, we seek to find the filtered estimation of the state i.e., the estimation of the state x_k based on the history of the past observation $y_{1:k}$ defined by $\{y_i : i = 1 \dots k\}$. Since x_k can be considered as a random variable, the characterization of x_k is given by the conditional probability density, $p(x_k | y_{1:k})$. In the Bayesian approach we follow, this can be obtained through recursive predictions and observations. Given an initial state, a *a priori* estimation of the state is calculated according to equation (5) in the prediction stage. After a new observation is taken a *a priori* estimation is updated to become a *a posteriori* state estimation.

By Bayesian filtering, we mean any technique that uses Bayes' formula to estimate the posterior density $p(x_k | y_{1:st_k})$ for each $k = 1, 2, \dots$. This usually consists of two important steps called *prediction* and *update*.

- i. (Bayesian Prediction) This step maps the previous *posterior density* $p(x_{k-1} | y_{1:st_{k-1}})$ into the one-step *prediction density* $p(x_k | y_{1:st_{k-1}})$. The formula for this step follows immediately from Bayes' formula and the Markov property of the system dynamics as

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1}. \quad (7)$$

- ii. (Bayesian Measurement Update) Now we want to use y_k and the Bayesian prediction $p(x_k|y_{1:st_{k-1}})$ from the first step to calculate the posterior density $p(x_k|y_{1:st_k})$. Then *a priori* density estimation at k will be updated according to a new observation y_k via Bayes' formula and using once again the Markov property of the system dynamics as the *posteriori* estimation

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{\int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k}. \quad (8)$$

Under the assumption of linearity, where system and observation dynamics are linear functions of x_k , the Kalman filter can be derived from the recursive relations (7) and (8). Due to the linearity the conditional probability density function $p(x_n|y_{1:n})$ becomes Gaussian and characterized by its mean and variance.

The power systems that we are interested in are nonlinear and the probability density functions are no longer Gaussian. As a remedy, an extended Kalman filter (**EKF**) is used in which the state and observation equations are linearized around the current state as a first approximation. In a strongly nonlinear environment these assumptions may undermine filter performance. Furthermore, the application of **EKF** to very high dimensional problems such as power systems, where the dimension of x and z for a typical Eastern U.S. power network can be of the order of 5,000 and 50,000 respectively, would be a computationally demanding problem even for state-of-art supercomputers. This is because the **EKF** needs to store and manipulate error covariance matrices, which becomes unmanageable for high dimensional models. In addition to this, the **EKF** assumes that the errors in the state estimates are small and that the state errors propagate through a separate linearized system. Because of this, **EKF** has a fundamental flaw since it is based on the attempt to approximate non-Gaussian density with the Gaussian one through a closure scheme, and this causes several nontrivial problems such as an unbounded error variance growth [4].

A recent, more efficient class of filtering methods is called particle methods, which includes sequential Monte Carlo, ensemble Kalman filters and interacting particle filters. Particle algorithms are techniques for implementing a recursive Bayesian filter by Monte Carlo simulations (see for example, Arulampalam et al [1]). Ensemble Kalman filters (**EnKF**) were proposed to resolve these defects in **EnKF** [4]. Approximating an error covariance matrix by an ensemble of states instead of storing a full matrix certainly reduces the storage demand. Replacing the linearized state evaluation in **EnKF** with integration of each member of the ensemble forward in time using Markov Chain Monte Carlo method made **EnKF** possible to handle nonlinear problems and succeed in many areas of application [5] since its development.

In a similar vein to **EnKF**, the particle filter approach uses an ensemble of states. But the ensemble is used to represent the probability density functions and the update from measurements is done by Bayes' rule (8) instead of the Kalman filter equation. For this reason, particle filters, in principle, are applicable to any nonlinear dynamical system. While there are many variants of particle filters, the basic algorithm underlying them is same. First, the prior density is

approximated by sample particles with weights. These particles are evolved according to system dynamics and once the new measurement y_k is obtained, new weights are assigned proportional to the likelihood of each prior sample. To avoid degeneracy of the samples and make sure the convergence of the algorithm, re-sampling is used if necessary. These steps are recursively done for each iteration.

In this work we will apply the extended Kalman filter for almost linear applications and the particle filters to the highly nonlinear system state estimation framework and compare their performance. Armed with these algorithms, we should be able to handle a large class of nonlinear filtering problems.

3 Software Tools

The heart of the work is the development of an overall system framework for a new approach to power system security monitoring. We are developing globally coordinated monitoring and decision making tools, and this requires global data sharing and coordination. The software tools and methodologies must be designed taking into account the physical power system dynamics and behavior in response to control actions and contingencies. The enabling computer science technologies to achieve this objective are Grid computing.

We are incorporating existing Grid computational tools and developing new tools, both to analyze and interpret electrical power grid state data and to integrate such data into power grid simulations to predict the future state of the system, as described in this paper. The data grid technology to be used in this project includes Web and Grid service technologies and standards, such as the Globus toolkit [6] and the Storage Resource Broker [2]. One way in which we are contributing to Grid computing technology lies in building effective methods for integrating power grid measurement data into both data analysis and simulations in a distributed computing and sensing environment.

The data grid work builds on our prior work on component-based tools for Grid environments, including DataCutter [3] and InterComm [8]. Much of this prior work has focused on providing high performance and flexibility in order to enable components to be effectively deployed onto distributed sets of computational resources. However, for this effort we are developing methods for providing performance guarantees. In the power grid monitoring and control environment, limiting response times is crucial. DataCutter is a system for enabling application developers to write simple functions (called filters) to perform parts of a data analysis application, connect the filters into a graph that performs all the desired processing of the data, and then deploy the filters onto remote computational resources. Standard Grid technologies are used for authentication across multiple sites and for starting up the required services. In addition, we have been building tools to index datasets stored across multiple sites [7], to enable efficient access for certain types of data requests (multidimensional range queries). InterComm is a programming model and runtime library for achieving direct data transfers between data structures managed by multiple parallel and sequential programs. Each program does not need to know in advance (i.e., before a data transfer is

desired) any information about the program on the other side of the transfer. All required information for the transfer is computed at runtime. InterComm allows a program to specify data distribution across processors in each program in a flexible manner. The InterComm framework also includes the infrastructure required to decouple the specification of when data transfers occur between coupled sets of components from the logic within the components. A separate coupling specification, written by the application integrator (the person building the coupled set of components), is used by the framework to specify when data transfers between coupled components should occur.

In this work, both InterComm and DataCutter services are being extended and enhanced to deal with the large, frequent data transfers required from multiple sources to parallel data analysis, control modules, and simulations. More specifically, InterComm services are being integrated with DataCutter services to deploy various components at distributed sites, and tie them all together to perform the required analyses. These services will also be used to couple the sensor data into the simulation models as it is produced and consumed, in a flexible manner. Much of the work also involve providing standard Web and Grid service interfaces, to enable the use of services developed in the wider Grid and Web communities, for security, authentication, information discovery, etc. The design and implementation of those services, including tools and interfaces that can be easily employed by power and control systems engineers, is a major challenge.

4 Automated Visualization for Power System Health Monitoring

Visualization techniques are being investigated to maximize the benefit of data streams and data content. Efforts to enable a more meaningful operator view include color contouring of wide-area data and 3-D animation. Examples of the data visualized are network bus voltages, line percent loadings, and area economic data. Displays are being tested under a variety of different simulated operating points to represent different emergency system conditions. The goal of this testing is to determine the displays that are best at helping users actually correct system problems. Our current hypothesis is that such visualization would be useful in helping people see subtle changes in key measures that might be important precursors to impending system problems. For operational use we envision the ability to "play back" animated visualizations showing the variation in the system conditions over a specified time period, similar to the animated weather radar images used to track storms. We will experiment with visualizing other system quantities, such as real power flow, and the hundreds of power transfers that simultaneously take place between the different control areas.

An example of such a data-driven visualization that we have performed is the contours of the per unit voltage magnitudes at several thousand 115/138 kV buses in the Ohio region for the simulated August 14, 2003 conditions immediately before the beginning of the events that led to the infamous blackout. Those contours provide an overview of the voltage profile of the entire region.

Hence the display is quite effective for providing situational awareness. For actual power systems this data has been traditionally supplied in real-time by the SCADA system with typical refresh rates varying between 5 and 30 seconds. In the near future, as PMUs become more ubiquitous, much faster refresh rates will be possible, allowing visualization of a system as it is moving toward voltage instability. The contours may show subtle oscillations in bus voltage magnitudes or angles.

Another possible advantage of using data driven, wide-area visualizations arises because power system operation is it tends to be cyclic, with strong a strong daily, weekly and seasonal cycles (e.g., very high loads on weekday afternoons during the summer, and much lower loads during the weekend nights in the spring or fall). Hence an experienced system operator might readily detect patterns that seem abnormal. For operational use one might envision the ability to "playback" animated visualizations showing the variation in the system conditions over a specified time period, similar to the animated weather radar images used to track storms. For human factors testing we might create animation sequences of different visualizations and test how readily the experiment participants detect something abnormal. We will also experiment with visualizing other system quantities, such as real power flow, and the hundreds of power transfers which simultaneously take place between the different control areas.

References

1. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, Vol.50(2), 174-188, 2002.
2. C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. In *Proceedings of CASCON'98 Conference*, Dec. 1998. <http://www.cas.ibm.com/archives/1998/index.html>.
3. M. Beynon, C. Chang, U. Çatalyürek, T. Kurc, A. Sussman, H. Andrade, R. Ferreira, and J. Saltz. Processing large-scale multidimensional data in parallel and distributed environments. *Parallel Computing*, 28(5):827-859, May 2002. Special issue on Data Intensive Computing.
4. G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.*, 99, 10, 14310,162, 1994.
5. G. Evensen. The Ensemble Kalman Filter: Theoretical formulation and practical implementation, *Ocean Dynamics*, 53, 343367, 2003.
6. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115-128, 1997.
7. B. Nam and A. Sussman. Spatial indexing of distributed multidimensional datasets. In *Proceedings of CCGrid2005: IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE Computer Society Press, May 2005.
8. J. S. Wu and A. Sussman. Flexible control of data transfers between parallel programs. In *Proceedings of the Fifth International Workshop on Grid Computing - GRID 2004*, pages 226-234. IEEE Computer Society Press, Nov. 2004.

Towards a Dynamic Data Driven System for Structural and Material Health Monitoring

C. Farhat¹, J.G. Michopoulos², F.K. Chang¹, L.J. Guibas¹, and A.J. Lew¹

¹ Dpt. of Mechanical Engineering and Computer Science
Stanford University Stanford, CA 94305, U.S.A.

{cfarhat, fkchang, guibas, lewa}@stanford.edu

² Special Projects Group, Code 6390.2

Center for Computational Material Science

Naval Research Laboratory, U.S.A.

john.michopoulos@nrl.navy.mil

Abstract. This paper outlines the initial motivations and implementation scope supporting a dynamic data driven application system for material and structural health monitoring as well as critical event prediction. The dynamic data driven paradigm is exploited to promote application advances, application measurement systems and methods, mathematical and statistical algorithms and finally systems software infrastructure relevant to this effort. These advances are intended to enable behavior monitoring and prediction as well as critical event avoidance on multiple time scales.

1 Introduction

During the last two decades, great strides have been achieved in many aspects of computational sciences and engineering. Higher-fidelity mathematical models, higher-order approximation methods, and faster solution algorithms have been developed for many applications. Computing speed and networking communications barriers have also been shattered by hardware manufacturers. As a result, the potential of modeling and simulation for reducing design-cycle time and enhancing system performance is recognized today in almost every field of engineering. However, for many complex structural systems, even the most elaborate computational models remain bound to be imperfect for many reasons. These include the limited means currently available for realistically modeling a number of phenomena governing the behavior of the system of interest or some of its components. They also include the fact that for most cases, stimulus conditions (initial, boundary, and loading) are typically acquired only after the actual systems are placed in service. This hinders the potential of modeling and simulation, and forces designers to impose empirical safety factors for allowable performance limits. This practice often leads to over-designed, inefficient and expensive products that may be unsafe in the presence of unforeseen critical conditions. The Dynamic Data Driven Application Systems (DDDAS) paradigm is uniquely opportune for exploiting maturing computational and sensor networking technologies to compensate for model deficiencies and unforeseen

system evolution and stimulus conditions, mitigate the effect of design imperfections on long-term as well as short-term system safety, and enable informed decision for maintenance planning and crisis management.

Our overall goal is to enable and promote active health monitoring, failure prediction, aging assessment, informed crisis management, and decision support for complex and degrading structural engineering systems based on dynamic-data-driven approaches. The specific objectives are: (i) To investigate, design, implement, illustrate and assess a DDDAS for health monitoring, failure prediction, and crisis management at various time-scales of degrading structural systems. (2) To build a test-bed and exercise it to guide the development of the desired DDDAS, and validate with experiments a representative set of its underlying approaches and methodologies. (iii) To exploit the above test-bed and design a series of educational experiments that can serve as pillars for educational exploitation of the DDDAS concept.

The research themes associated with these specific objectives are driven by the following representative application scenarios: (S1) Apply the DDDAS concept off-line to obtain material characterizations and construct an initial model of the structural system of interest (S2) Exploit dynamic operational data to update regularly, on-line, the current model. (S3) Apply the DDDAS concept to perform on-line crisis management. (S4) Apply the DDDAS concept to assist off-line crisis management and maintenance planning.

Our project will draw experiences and technologies from our previous efforts on a data-driven environment for multiphysics applications (DDEMA) [1].

2 Approach

Scenario S1 outlined above is representative of the problem of collecting and exploiting data to develop a behavioral model for a system prior to its usage. Scenarios S2 to S4 cover a time-scale that ranges from very short as in catastrophic failure induced (e.g. by flutter manifesting in less than 2 seconds), to very long as in a failure induced (e.g. by a slowly accumulating damage due to cyclic load manifesting in months or years of platform operation).

Figure 1 captures the various application scenarios of the envisioned DDDAS and displays a road map of the corresponding research activities as follows. Material characterization will provide behavior models to the damage detection schemes (path 1); on-line processing and prognosis procedures will be updated by sensor-originating dynamic data (path 2); in turn, these procedures will modify the damage detection schemes (path 3) which will either provide feedback directly to the system if an imminent crisis is detected (path 4); otherwise, they will update off-line high performance modeling and simulation procedures and request long-term prognosis of slow evolving but accumulating damage (path 5). This can then enable on-demand rather than routinely scheduled maintenance.

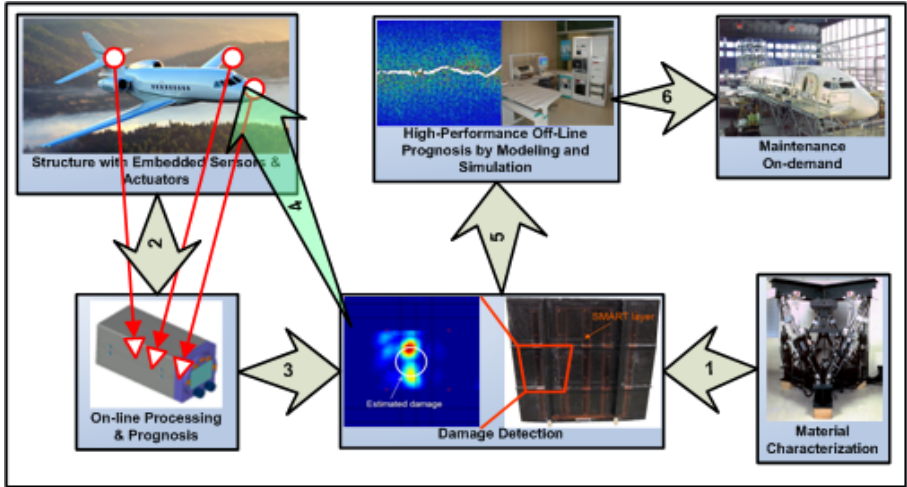


Fig. 1. Data flow within the main activities associated with our approach

2.1 Dynamic-Data-Driven Inverse Material State Characterization

An essential element of our approach is the data-driven computation of a constitutive theory/model that accounts for multiphysics loading conditions. This approach consists of the following parts.

A. Mechatronically Automated Data Acquisition. Data-streams of experimentally controlled continuous system behavior in the form of stimulus-response pairs of all observables will be generated. This will be done with the aid of an extended multi-degree of freedom mechatronic loading systems [2, 3]. While the system imposes preprogrammed displacement controlled loading paths to the appropriated specimens, sensors measure both the controlled displacements and rotations as well as the corresponding reaction forces and moments along with the additional multi-field state variables (i.e. temperature). These sensors generate the data-streams upon which all subsequent energy based evaluations are made. Of particular interest to DDDAS in general, is the opportunity we have to use our system in a manner which demonstrates that the modeled behavior can be used to adjust an experimental procedure.

B. Inverse Modeling of Dissipative Behavior. Analytical representations will be constructed such that they can accurately reproduce a subset of the acquired data via successive dynamic adaptation with the help of optimization technology. This will be accomplished by the employment of an energy density dissipation potential called dissipated energy density function (DEDF) that is a function of the strain and temperature states of the material. Aerodynamically induced heat can further contribute to the damage behavior of aerospace composite structures and therefore, the coupled influence of temperature with mechanical deformations cannot be neglected any longer and will be integrated in the formalism. The DEDF corresponds to the energy lost into the material

system for the creation of micro-cracking damage and therefore it represents a scalar physical quantity associated with material health [2, 4, 5]. Its gradient with respect to strains yields the corresponding constitutive relations that determine the stresses. An approximation of the DEDF for a material system will be constructed as a sum of appropriately selected basis functions that depend only on the local strain and temperature state of the material in the structure. The coefficients that are weighting these basis functions are the unknowns to be determined with the aid of the experimental data. Minimizing the error between the analytically computed dissipated energy and the experimentally measured one by NRLs mechatronic systems reduces this problem to a data-driven inverse global optimization problem with appropriate inequality constraints. Solving this problem by any global optimization technique produces the sought-after coefficients and fixes the form of the DEDF.

C. Degrading Behavior Simulation. The derived constitutive models derived will be utilized to simulate the response of any system that uses the characterized materials. To achieve this, the identified degrading behavior model will be introduced into our AERO-S structural code [6, 7]. AERO-Ss separation of material and geometric encoding makes this effort very feasible as we have already demonstrated for single physics applications [2, 3, 4, 5].

D. On-line Extensions. The material characterization procedures described above are essentially off-line activities. Online exploitation of damage accumulation models, damage tracking criteria and algorithms will also be developed based on the amount and rate of dissipated energy in a fashion similar to the one developed for metal fatigue.

2.2 Sensor Network Architectures and Protocols

Any prediction based on the approach outlined above depends, among others, on the performance of the sensor network providing the data. In the context of coupling the simulation of a structure to a sensor network monitoring the structure, some problems take on a special significance and need to be addressed.

On-line dynamic techniques that monitor the health of the routing infrastructure of the network, bypass failed areas, and always guarantee information delivery as long as the network remains connected will be investigated. We already have developed distributed techniques for discovering failed areas, or “holes”, in a sensor network and building routes around them [8].

Regarding the data integrity we intend to investigate how to tune sensor misbehavior detection system, as too many false positives may remove valuable sensor resources from the network, while too many false negatives may pollute the data generated and veer the simulation off track. Unlike other sensor network settings, the availability of a powerful simulator in our scenarios can greatly aid the discovery and correction of node misbehavior.

Relative to the coupling between sensors and simulation technology, our goal is to define a narrow interface between the sensor network and the simulator, exploiting our knowledge of models of how the system should behave in order to compress away predictable information and focus only on the deviations between

the simulation predictions and the actual readings. We are constrained by the fact that the nodes have very limited computational power, so it is not possible for them, individually or in groups, to run the full simulation model.

To achieve this goal, an approach where the simulation computes reduced-order models that are appropriate for making predictions over a certain time period for small groups of collocated sensors (sensor clusters) will be followed. These models are transmitted to the sensor clusters and then independently checked by distributed regression within the cluster. No communication with the base station is necessary as long as the local models fit the data. The communication between the sensor network and the simulation is reduced to the periodic transmission of reduced-order models from the simulation and of deviant sensor readings from the nodes. Even if reduced-order models are difficult to obtain for some applications, general data aggregation and summarization techniques can be exploited to the same effect, though not with the same efficiency [9]. These approaches can be used to define narrow interfaces between the simulation and the sensor network and can form the basis of a software infrastructure that serves well data-driven application systems.

2.3 Modeling Approaches and Solution Algorithms

The application problems targeted by this research effort are not linear time-invariant. Hence, constructing reduced order models (ROMs) can be challenging. Some nonlinear model order reduction techniques have already been developed and encountered some success. However, these techniques are in general either applicable only to weakly nonlinear systems, or they are reliable only for input signals close to the training input. Furthermore, they are not usually robust with respect to large parameter changes, whether the parameter is geometrical, material, or related to the surrounding of the structural system for example, the Mach number of a cruising aerospace vehicle [10]. Reconstructing a ROM each time a problem parameter is varied cannot be performed today anywhere close to near real-time, and therefore defeats the purpose of using a ROM in the first place. Alternatively, a ROM can be rapidly adapted to a parameter change. Previous approaches for adapting a proper orthogonal decomposition (POD) basis to address a parameter variation include the global POD method [11, 12] and the direct interpolation of the basis vectors [10] with poor results.

Two approaches for adapting a ROM will be investigated. The first approach aims at short time-response or on-line predictions where changes in the material properties of the structure slightly before the onset of the short-term instability and a couple of seconds later can be neglected. The objective is to rapidly adapt a ROM trained at a particular set of stimulus conditions to another set of loading conditions. For such applications, a suitable combination of a sensitivity approach and the method of “subspace angle interpolation” [13] will be investigated. We expect that the combination of this approach with sensitivity analysis and a precomputing strategy will lead to an effective approach for adapting in near-real-time a given ROM to changes in multiple parameters. The second approach aims at off-line medium time-response problems where fast but

not necessarily near-real-time processing is required. It is based on the re-use of Krylov subspaces for the solution of near-by problems [14].

We have recently developed a family of Parallel Implicit Time-integration Algorithms (PITAs) for accelerating the numerical processing of ROMs in order to achieve near-real-time predictions [15, 16]. In [16], it was shown that for linear but otherwise complex structural dynamics problems, the latest PITAs can converge in two to three iterations and speed-up the total processing of a ROM by a factor equal to six using 25 processors. Hence, the objective of a PITA is not to deliver a high parallel efficiency, but to reduce as much as possible the CPU time of a problem that offers little opportunities for parallelism. We are attempting to extend the PITA methodology to nonlinear structural dynamics problems. Furthermore, in order to get closer to near-real-time response, we are also attempting to investigate two complementary approaches for improving the CPU performance of the PITA methodology. The first approach consists in investigating an alternative to propagating the jumps of the solution on the coarse time-grid in order to reduce the computational overhead of this time-parallel methodology. The second approach consists in exploiting the DDDAS context and initializing the seed values of the solution by sensor-provided data.

2.4 Systems Software Technology

Figure 2 presents an abstract architecture that has both off-line and on-line functionality and remains consistent with the tenets of our project. It consists of the following main components: a sensor subsystem, a narrow interface between the sensor subsystem and the sensor data reduction and/or preparation subsystem, a user interface, a full order model (FOM) solver with the ability to construct ROMs, a ROM database, a ROM adaptor that composes appropriate on-line models based on sensor inputs, and a behavior visualizer. This architecture is capable of addressing both the off-line and on-line modes of operations as they were described earlier. In particular, the path defined by the sequence

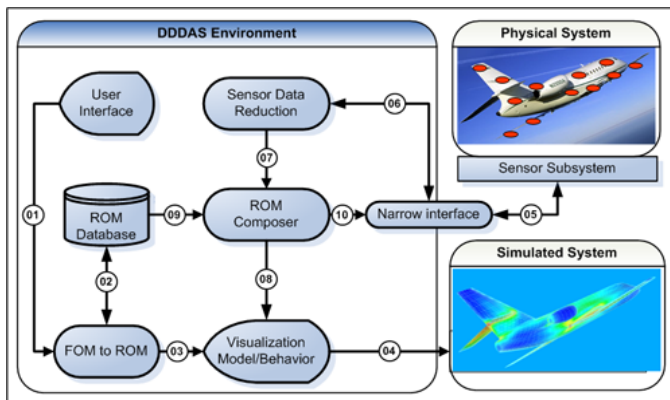


Fig. 2. Generic DDDAS architecture

[01,02,03,04] in Figure 2 can be executed off-line and can be used for computing and storing ROMs. Similarly, the path [05,06,07&09,08,04] can be used for ROM-based and sensor-driven simulations for decision support. Finally, the path [09,10,05] implements the model-based effects on the sensor network. Some software components in this architecture will dynamically select and optimize their distribution at run-time via mobile “middleware” as shown for DDEMA [1].

2.5 Test-Bed Article Integration and Demonstration

A general test-bed infrastructure of both wired and wireless sensor networks will be selected for building the prototype in the study. For a wired network system, the “SMART” layer approach [17] will be adopted. Passive sensors such as strain and temp sensors will be embedded along with active piezoelectric sensors/actuators into the layer. The layer will be customized fit into the configuration of the wing, which will be made of graphite epoxy prepreg commonly used in aircraft structures. Sensor network strips will be embedded inside composites or mounted on the inner surface of the wing during curing process as shown in Figure below so the composite and layer were integrated during the fabrication.

For the study of short-term response, the composite wing will be subjected to a foreign object impact by a hammer or a drop weight at the structures. Upon impact, the same sensor networks will be activated automatically to characterize the impact event and determine if active sensing is needed to interrogate the severity of the impact damage to create diagnostic images. The estimated damage image will be imported into a finite element simulation code to determine the effect of such damage on the residual strengths of the structure.

Finally, for the study of long-term time scale, the composite wing will be subjected to repetitive flight conditions determined by repetitive combinations of angle of attack and speed in order to induce and to assess the long-term degradation. Sensors will be reporting strains and temperature data for a multiple missions over long period of time. Tracking of the strain-temperature condition will be accomplished by appropriate database utilization along with corresponding dissipated energy density evaluations. After each emulated mission is completed, the final distribution of dissipated energy density will be assessed and used to derive the current softened distributions of the material properties (at the areas where this is applicable), to be used as startup conditions for the next emulated flight. This process will be repeated for various emulated flights and the computed material softening distributions as a result of slowly accumulating damage will be compared with the correlated values of damage index evaluated from the short-term (or medium time) health monitoring technique that will be employed.

3 Conclusions

In this paper, we have described an outline for the motivational drivers, some technical issues and the approach for developing a DDDAS for material and

structural health monitoring along with critical event prediction across heterogeneous time-scales. Both the main research activities of the project and their interrelationship as well as an abstract architecture of the system have been described. The general application area for such a system is the area of design and maintenance of aerospace platforms.

Acknowledgement. The authors acknowledge the support by the National Science Foundation under grants EIA-0205663 and CNS-0540419. Partial support from NRL's 6.1 core-funding is also greatly acknowledged.

References

1. Michopoulos, J., Tsompanopoulou, P., Houstis, E., Farhat, C., Lesoinne, M., Rice, J., Joshi, A., On a Data Driven Environment for Multiphysics Applications, *Fut. Generation Comp. Sys.*, **21**(6), (2005), 953–968.
2. Mast, P. W., Michopoulos, J. G., Thomas, R. W., Badaliance, R., and Wolock I., Characterization of strain-induced damage in composites based on the dissipated energy density: Part I - basic scheme and formulation. *Int. Jnl. of Theor. and Applied Fract. Mech.*, **22**, (1995), 71-96.
3. Michopoulos, J., Mechatronically automated characterization of material constitutive response. in: *Proc. of the 6th World Congress on Computational Mechanics (WCCM-VI)*, (2004), 486-491.
4. Michopoulos, J. G., Mast, P. W., Badaliance, R., and Wolock I., Health monitoring of smart structures by the use of dissipated energy. in: *Proc. 93 WAM on Adaptive structures and material systems*, G.P. Carman and E. Garcia, editors, ASME, vol. ASME, AD-Vol **35**, (1993), 457-462.
5. Factory, P. R., Health error prediction and sensor topology optimization on a smart pressure vessel. In *SPIE Proc. Smart Structures and Materials 1995, Industrial and Commercial Applications of Smart Structures Technologies*, volume SPIE **2447**, (1995), 155-166.
6. Geuzaine, P., Brown, G., and Farhat, C., Three-field based nonlinear aeroelastic simulation technology: Status and application to the flutter analysis of an f-16 configuration. in: *Proc. of 40th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2002-0870*, (2002), 14-17.
7. Farhat, C., Geuzaine, P., and Brown G., Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter. *Computer and Fluids*, **32**, (2003), 3-29.
8. Fang, Q., Gao, J., and Guibas, L., Locating and bypassing routing holes in sensor networks. in *23rd Conference of the IEEE Communications Society*, (2004).
9. Shrivastava, N., Buragohain, C., Agrawal, D., and Suri, S., Medians and beyond: new aggregation techniques for sensor networks. in *Proc. ACM Conf. on Embedded Networked Sensor Systems*, (2004), 239-249.
10. Lieu, T., and Lesoinne, M., Parameter adaptation of reduced order models for three-dimensional flutter analysis. Number AIAA Paper 2004–0888, (2004).
11. Taylor J., and Glauser, M., Towards practical flow sensing and control via pod and lse based low-dimensional tools, in *Proc. of ASME Fluids Engineering Division Summer Meeting, FEDSM2002-31416*, Canada, (2002).
12. R. Schmidt and M. Glauser. Improvements in low dimensional tools for flow-structure interaction problems: using global pod. Number AIAA Paper 2004–0889, (2004).

13. Lieu, T., Farhat, C., and Lesoinne, M., Pod-based aeroelastic analysis of a complete f-16 configuration: Rom adaptation and demonstration. In 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conf., AIAA Paper No. 2005-2295.
14. C. Farhat, K. Pierson, and M. Lesoinne. The second generation of feti methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. *Comp. Meth. in Appl. Mech. and Engng.*, 184:333-374, (2000).
15. Farhat, C., and Chandesris, M., Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid-structure applications. *International Journal for Numerical Methods in Engineering*, **58**, (2003), 1397-1434.
16. Cortial J., and Farhat, C. A time-parallel implicit methodology for the near-real-time solution of systems of linear oscillators. (in press).
17. Ihn, J. B.. and Chang, F.K., A smart patch for monitoring crack growth in metallic structures underneath bonded composite repair patches, in: *Proc. of the American Society for Composites 17th Tech. Conf.*, Purdue University, (2002).

The Omni Macroprogramming Environment for Sensor Networks

Asad Awan, Ahmed Sameh, and Ananth Grama

Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907

Abstract. Structural sensing and control is an important application of the DDDAS paradigm. Our work on structural sensing and control has several key aspects, including model reduction, control, simulation, and validation. Motivated by our work on validation using an actual three-storeyed structure, we are developing a comprehensive systems environment, Omni, for macroprogramming sensor networks. While there have been efforts targeted at enabling programmers to write lean applications for individual sensor nodes, there have been few efforts targeted towards allowing programmers to program entire networks as distributed ensembles. Omni provides an intuitive and efficient programming interface, along with operating system services for mapping these abstractions into the underlying network. In this paper, we provide a high-level overview of the Omni architecture, its salient features, and implementation details. The Omni architecture is designed to be a flexible, extensible, scalable, and portable system, upon which a wide variety of DDDAS applications can be built.

1 Introduction and Motivation

An important application of the Dynamic Data-Driven Application Systems (DDDAS) paradigm is in the control of large civil structures. It is estimated that the United States has an investment of over \$20 trillion in its civil infrastructure. While these systems are in constant cycle of deterioration and renewal, they are expected to withstand extreme loads caused by natural disasters as well as human factors. Protecting this investment is of prime social and economic importance. The serviceability and safety of these systems, ranging from high-rise structures and long-span bridges to major pipelines that traverse the U.S., can be greatly improved if damage can be detected and controlled intelligently before catastrophic failures. The technology for detection and control of damage is, in principle, now available through low-power sensors, actuators, and communication and computing elements. As part of our NSF-funded project (ITR/DDDAS), we are developing the necessary computational infrastructure to enable: (a) the effective design and economical construction of highly robust smart structures capable of significantly greater resilience to catastrophic events; (b) enhancing robustness of existing structures by suitably retrofitting smart sensor-actuator complexes; (c) predicting and mitigating the impact of catastrophic events in real time; and (d) off-line data archival and analysis technologies for establishing failure causalities and design enhancements.

We identify several intellectual challenges in the eventual application of closed loop control to structures. We specifically target algorithms and techniques for model reduction and control, analysis and simulation, and systems infrastructure for real-time sensing and actuation. As part of our prior work, we have developed extensive algorithmic

and software infrastructure for model reduction and control [16, 10, 8, 6, 9, 2, 3, 5, 4, 14], simulation, and visualization [11, 12, 13]. Our work on simulation and visualization resulted in the first comprehensive science-based analysis of the Pentagon crash. Subsequent efforts have modeled heat-induced structural failure in high-rise buildings. Our current efforts in this direction are aimed at a comprehensive modeling of the structural failure of the World Trade Center. Our modeling and simulation work is complemented by experimental tests at the Bowen Lab of Structural Engineering at Purdue. This unique facility provides us with a powerful validation mechanism for our computational approaches.

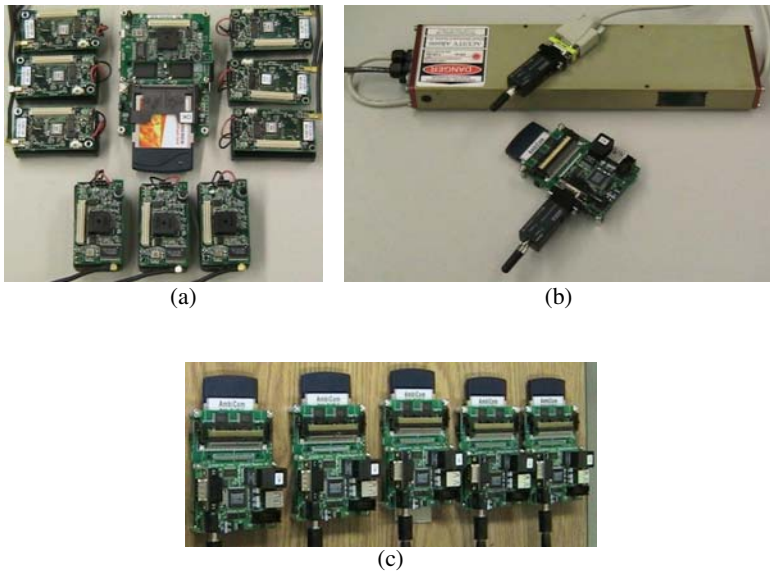


Fig. 1. Some elements of our current sensing infrastructure: (a) Mica 2 motes with 8MHz Atmega128 μ c, ADXL202 Micro-electro-mechanical (MEMS) 2-D accelerometer (range $\pm 2g$, sensitivity 12.5%/g, 2mg resolution at 60Hz, power $< 0.6mA$ at $\sim 3V$); (b) laser displacement sensor, Acuity AR600, accuracy up to 0.04 in, sampling rate up to 1.25KHz; and (c) communication and in-network processing hubs with 400 MHz XScale processors, and communication interfaces for FM, Bluetooth, and 802.11b

As part of these validation efforts, we have built a structural sensing infrastructure based on accelerometers, high-accuracy laser displacement sensors, strain gages, and compute and communication elements (Figure 1). In building this sensing infrastructure, we realized the critical need for a systems environment for (macro)programming sensor networks. This macroprogramming environment is designed for real-time sensing as well as affecting control. It provides support for essential characteristics, including robustness, reconfigurability, and real-time guarantees. In the rest of this paper, we describe the architecture, programming model, and current status of our development efforts towards the Omni environment.



Fig. 2. An instrumented test infrastructure at the Bowen Labs (three story $30' \times 50'$ structure) with extensive actuation and response testing capability

2 Overview of the Omni Macroprogramming Environment

An operating system architecture defines the design paradigm, and hence behavior and performance, of the applications built on top of it. We present a second generation sensor network operating system suite, Omni, which facilitates rapid development of efficient self-organized distributed applications for sensor networks. Existing sensor network operating systems (for example [15, 7, 1]) allow the development of network-enabled applications at each node, however, their design does not directly facilitate the role of application components in the *distributed* behavior of the sensor network as a whole. While these systems address the problem of developing lean (low CPU and memory overhead) applications, they are apathetic to the complexity of designing and implementing high performance distributed functionality in sensor networks. This is a critical requirement for the development of large-scale sensing and control applications.

Other approaches for designing and implementing distributed behavior over sensor networks include the use of domain specific programming languages (for example [20, 19]). However, the use of high level languages often abstracts away low-level, system dependent details, which often need to be tuned, given the resource constraints of sensor nodes. Finally, stream processing database systems for sensor networks ([18, 17]), allow SQL-type queries and aggregation operations over data streams. While this approach affords ease and flexibility, the performance on (typically) resource constrained nodes is limited due to the generality of the database stream management system (DSMS), and customization is limited by the complexity of the design and implementation of the system.

Apart from aiding the design and development of distributed applications, Omni provides several other key features. Similar to SOS [7], we support dynamic update of application components and OS services. However, the distributed design paradigm supported by our system directly allows network wide self-organized adaptation rather than the updation of each module treated independently. Our application development paradigm supports concurrency safety between application components, executing on the same node, irrespective of the underlying scheduling policy or the number of

on-board processing elements. The stream processing paradigm allows a programmer to view processing in each component of the application as an independent transaction and relieves the programmer from the burden of maintaining synchronization in a distributed system. Similarly, the stream processing abstraction is ideally suited to the ad-hoc peer-to-peer model of sensor network systems in contrast with client-server based abstractions such as RPC.

3 Omni Architecture

The design of the Omni OS focuses on providing abstractions that allow rapid development of robust and efficient distributed applications for sensor network systems. Our design separates the core OS kernel, OS services, and distributed applications. The core kernel and the OS services control the behavior of a single node, while applications implement the distributed system behavior of the sensor network. Following the stream processing model the applications are developed using a box-and-arc model. Each box represents a *processing element* and arcs represent data channels. However, unlike most systems that are based on this model, we allow runtime reconfiguration of the model, including changing interconnections and replacement of processing elements (PEs). The applications view the sensor network as a single large system, albeit distributed, allowing simplified design. Each processing element is conceptually a single execution unit interfacing only via the input and output channels. A PE can not request dynamic memory except for the dynamically growing input and output queues. In practice, the isolation of a PE provides strong concurrency and memory safety properties. The channels provide transparent and efficient transportation of data streams between PEs in the distributed space (i.e., communicating PEs can be located on the same, or different physical nodes). Channels are implemented as low-overhead lock-free single producer, multiple consumer (SPMC) queues mitigating communication overheads.

3.1 Omni Processing Elements

The design specification of a PE only declares its *functional* behavior and *typed* input and output streams. Therefore, at design time, a PE's relationship with other PEs, or its placement in an application (fundamentally, which is a given instance of the box-and-arc diagram) is not known. The separation of the design of a PE from application design enforces modularization and allows code reuse. Once a repository of functionally specific PEs is developed, a new application can be developed rapidly by designing a connection diagram and providing complementary specifications. These diagrams can be statically checked, e.g., for typed-correctness, to verify that the application meets the distributed behavioral specification. Static checking also removes the burden of verifying interfaces at runtime on resource constrained nodes. In existing sensor network OS examples [7], limited verification is performed at runtime, which affects the safety properties of the system. The expected complexity of interactions between application components further burdens the programmer and makes debugging difficult.

Conceptually a PE is treated as a single unit of execution. At runtime this abstraction is enforced by treating each execution of a PE as a transaction. Each transaction reads

input streams, generates output streams, and on success requests the system to commit. A commit involves removing the consumed data from the input streams and making the output streams available for downstream processing. Inability of the OS to perform the commit results in no change to the input or the outputs. This is notified to the PE and a re-execution of the entire transaction follows. This model also allows the system to preempt or *kill* a currently executing PE without generating any inconsistency.

The processing elements are implemented using C language and are compiled using platform specific gcc compiler and linker scripts (to enable dynamic runtime loading). This allows seamless portability of the PEs requiring only recompilation for different platforms. The application is a specification expressed as a schema and is processed by our custom compiler, which in general is platform agnostic, but may be modified to account for optimizations or irreconcilable differences between target platforms. In the future, application design may be generated by a WYSIWYG (what you see, is what you get) GUI front end that generates the schema.

3.2 Omni OS Services

The OS services are designed using the same box-and-arc model, thus are similar to services in a micro-kernel based design. Here, the key difference is that a single multiplexed queue is replaced by point-to-point channels. The connection between services is determined via the box-and-arc configuration rather than use of addresses (which, in practice, are often hard-coded) as in most micro-kernel architectures. In contrast to an application PE, OS services are designed to control the underlying hardware and behavior of each node independently. The operational similarity between PEs and OS services allows a uniform OS runtime architecture, while the design time specifications

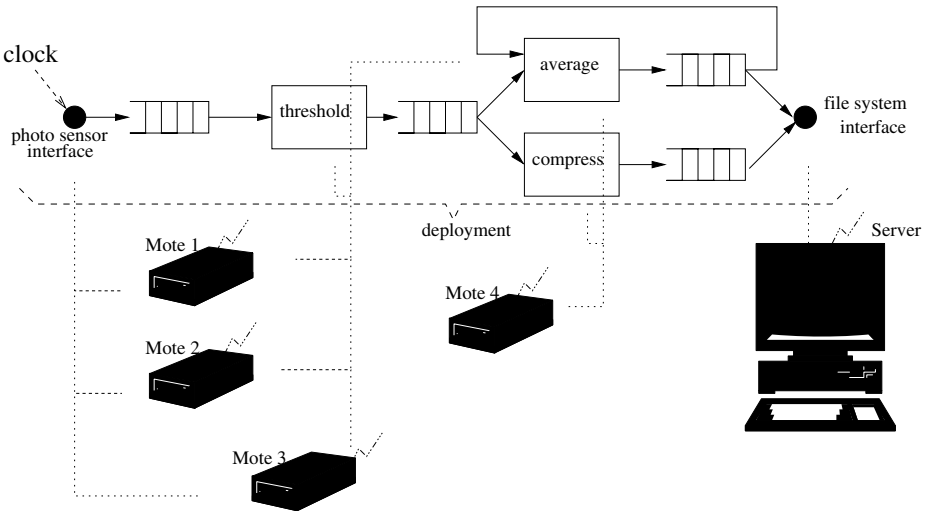


Fig. 3. An example stream processing application connection graph and its deployment over a sensor network system

```

/* ----- A basic application schema ----- */
@ PHOTO_SENSOR_NODE:
TRIGGER{Clock, RATE}          -> ADC[PHOTO_SENSOR]

@ NODE_ANY:
ADC[PHOTO_SENSOR]             -> pe_threshold(s_photo_t in)
pe_threshold(s_photo_t out)   -> pe_average(s_photo_t in_p)
pe_average(s_avg_t out)       -> pe_average(s_photo_t in_avg+)

@ NODE_FAST_CPUS:
pe_threshold(s_photo_t out)   -> pe_compress(s_photo_t in_p)

@ NODE_SERVERS
pe_compress(s_comp_t out_c)   -> FILE[COMPRESS]
pe_average(s_avg_t out)       -> FILE[AVG]

/* ----- Threshold function ----- */
int pe_threshold(flag_t *in_flag, flag_t *out_flag,
                /*in*/ s_photo_t *in, /*out*/ s_photo_t *out)
{
    stream_t *ins, *outs;

    ins = get_data_stream(in);
    outs = get_data_stream(out);

    while(has_data(ins))
        if (get_data_val(ins) > THRESH)
            set_data(outs, ins);

    commit(out_flag, in );
    commit(out_flag, out);
    return OK;
}

```

Fig. 4. Code sample illustrating the application schema (for the example in Figure 3) and the C Language implementation of the threshold PE

and compile time enforcement maintain the conceptual differences. Note that at the application level, OS services and the core OS are invisible in the application connection diagram. For example, an OS service implementing the routing service is abstracted away by the distributed channel model used by the application PEs. However, the developer still has control over the low-level system components, for example in this case the network implementation of the channel, due to the ability to modify the OS services to monitor and manage system hardware. At the platform level, the core OS kernel provides a hardware abstraction layer (HAL) decoupling the OS services from the platform hardware. Thus, the system design and development is cleanly partitioned into application, OS services, and core OS components.

The OS services and applications (including individual PEs) can be reconfigured and updated at runtime to adapt the performance and functionality of a single node or the entire sensor network system, respectively. The OS also has the capability to autonomously reconfigure the components based on design time instructions and specifications. Provision for terminating an executing PE or OS service at runtime allows updates without sacrificing consistency of the data stream (due to the transaction abstraction).

An example application is illustrated in Figure 3. A connection diagram with different processing elements is shown at the top. At the bottom the deployment of this conceptual connection diagram over the network is illustrated. Different processing elements are assigned to different nodes based on an application schema, shown in

Figure 4). The deployed processing elements are transparently connected by channels in the distributed space. Figure 4 also provides an example implementation of the threshold PE in C language.

4 Status of Development Efforts

We are currently developing and exhaustively testing the Omni environment targeted for the AVR (e.g., Mica2 mote) and POSIX platforms. We are also developing utilities for static checking, schema compilation, and over-the-network PE/OS service update tools. We will port the HAL to other sensor node platforms, allowing reuse of the PE, OS service and most core OS components.

The Omni architecture provides a flexible, portable, and scalable platform over which a number of DDDAS applications can be built. These include data acquisition and sensing, control, as well as analyses applications. We aim to release the entire Omni environment after comprehensive validation, along with exemplar applications (from structural sensing), over the public domain.

References

1. Mate: Programming Sensor Networks with Application Specific Virtual Machines. <http://www.cs.berkeley.edu/~pal/mate-web/>.
2. P.A. Absil, R. Sepulchre, P. Van Dooren, and R. Mahony. Cubically convergent iterations for invariant subspace computation. *SIAM J. Matrix Anal. Appl.*, 2003.
3. Y. Chahlaoui and P. Van Dooren. Benchmark examples for model reduction of linear time invariant dynamical systems. *Model Reduction of Dynamical Systems, Eds. P. Benner et al.*, 2004.
4. Y. Chahlaoui and P. Van Dooren. Model reduction of time-varying systems. *Model Reduction of Dynamical Systems, Eds. P. Benner et al.*, 2004.
5. Y. Chahlaoui, K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of second order systems. *Model Reduction of Dynamical Systems, Eds. P. Benner et al.*, 2004.
6. Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren. Second-order balanced truncation. *Lin. Alg. Appl.*, 2003.
7. Chih-Chieh Han and Ram Kumar Rengaswamy and Roy Shea and Eddie Kohler and Mani Srivastava. SOS: A Dynamic Operating System for Sensor Networks. In *Proceedings of the Third International Conference on Mobile Systems, Applications, And Services (Mobisys 05)*, 2005.
8. P. Van Dooren. The basics of developing numerical algorithms. *Control Systems Magazine*, 18-27, 2004.
9. K. Gallivan, X. Rao, and P. Van Dooren. Singular riccati equations stabilizing large-scale systems. *Lin. Alg. Appl.*, 2003.
10. Y. Hachez and P. Van Dooren. Elliptic and hyperbolic quadratic eigenvalue problems and associated distance problems. *Lin. Alg. Appl.*, 371:31-44, 2003.
11. C. Hoffmann, S. Kilic, V. Popescu, and M. Sozen. Integrating modeling, visualization and simulation. *IEEE Computing in Science and Engineering*, January/February 2004.
12. C. Hoffmann, S. Meador, S. Kilic, V. Popescu, and M. Sozen. Producing high-quality visualizations of large-scale simulations. *IEEE Visualization*, 2003.

13. C. Hoffmann and V. Popescu. Fidelity in visualizing large-scale simulations. *Computer-Aided Design*, 2006. To appear.
14. B.F. Spencer Jr., S.J. Dyke, M.K. Sain, and J.D. Carlson. Phenomenological model of a magnetorheological damper. *ASCE Journal of Engineering Mechanics*, 2006. To Appear.
15. Philip Levis, Sam Madden, David Gay, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. TinyOS: An Operating System for Sensor Networks. *Ambient Intelligence*.
16. W.H. Liao and C.Y. Lai. Harmonic analysis of a magnetorheological damper for vibration control. *Smart Mater. Struct.*, 11:288-296, 2003.
17. Samuel Madden, Michael Franklin, Joseph Hellerstein, and Wei Hong. TinyDB: An Acquisitional Query Processing System for Sensor Networks. In *Proceedings of TODS*, 2005.
18. Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *Proceedings of OSDI'02*, December 2002.
19. Ryan Newton, Arvind, and Matt Welsh. Building up to Macroprogramming: An Intermediate Language for Sensor Networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*, April 2005.
20. Ryan Newton and Matt Welsh. Region Streams: Functional Macroprogramming for Sensor Networks. In *Proceedings of the First International Workshop on Data Management for Sensor Networks (DMSN)*, August 2004.

Evaluation of Fluid-Thermal Systems by Dynamic Data Driven Application Systems

D. Knight, T. Rossman, and Y. Jaluria

Dept of Mechanical and Aerospace Engineering
Rutgers - The State University of New Jersey
New Brunswick, NJ 08903
knight@soemail.rutgers.edu

Abstract. A Dynamic Data Driven Application Systems (DDAS) approach is developed for evaluation of fluid-thermal systems wherein a complete specification of the boundary conditions is not known *a priori* and experimental diagnostics are restricted to a limited region of the flowfield. The methodology is applied to the configuration of a heated jet injected into a laminar boundary layer where the jet temperature is not known *a priori*. Preliminary results are presented.

1 Introduction

The design and control of efficient fluid-thermal systems requires detailed quantitative information on the flowfield (*e.g.*, pressure, species concentration, temperature and velocity). Examples of such systems are combustors, furnaces, and reactors [1]. In many situations, very limited access to the flow domain is available and strong constraints are thus placed on the experimental data that can be obtained through non-intrusive (*e.g.*, optical) and intrusive techniques. An example is the optical fiber drawing furnace, which typically has an infrared sensor to monitor the temperature of the heating element at only *one* location and must use this single source of information to control the optical fibre drawing process [2, 3, 4]. Another example is the combustor in a turbofan engine.

A common approach is to use numerical simulation (*i.e.*, Computational Fluid Dynamics [CFD]) to obtain the desired information in the given domain using the available experimental data. However, this procedure can be successful *only* if the boundary conditions for the numerical simulation are well defined. In many cases of practical interest, such as the draw furnace and combustion chamber mentioned previously, the boundary conditions (in particular, the thermal conditions) are not known with sufficient accuracy or even at all. In the optical fibre drawing furnace, for example, the temperature distribution at the wall is the result of the overall heat transfer processes in the domain and cannot be taken as a known input to the model. This temperature distribution affects the fluid properties, as well as chemical reactions and the transport mechanisms in the flow. The wall temperature distribution is critical to the calculation of the glass flow and thus the characteristics of the fiber. However, it is typically infeasible to measure the wall temperature, and hence the information regarding the

boundary conditions is incomplete. It is possible to infer the boundary conditions from an inverse procedure. For example, Issa *et al* [2] determined the temperature distribution at the wall of a cylindrical fiber drawing furnace by using the limited experimental data obtained by measuring the temperature at the axis of a graphite rod located at the center of the furnace using an inverse calculation based on the governing transport equations coupled with an optimization procedure.

2 Dynamic Data Driven Application Systems

The Dynamic Data Driven Applications Systems (DDDAS) concept was described in the DDDAS Workshop held at the National Science Foundation in March 2000 [5]. DDDAS is a novel approach to engineering and scientific research wherein experiment and simulation interact in a synergistic, symbiotic manner. DDDAS can be implemented in either an open loop or closed loop manner. In the former case, experimental data is streamed into the simulation (or vice-versa) to achieve greater accuracy, detail and/or robustness. An example is real time streaming of meteorological measurements into weather simulations. In the latter case, experiment and simulation interact iteratively, where the experiment guides the simulation, and the simulation in turn guides the experiment (see Darema [6]). The recent Dynamic Data Driven Applications Systems Workshop in January 2006 [5] summarized major achievements using the DDDAS approach and identified additional opportunities. Additional information is available on the DDDAS.org homepage [7].

The Dynamic Data Driven Applications Systems concept may be directly applied to significantly improve the evaluation of fluid-thermal systems using combined experiment and simulation. The basic concept is to synergize the experiment and simulation in a manner which exploits the advantages, and recognizes the limitations, of each approach. We specifically focus on systems evaluation in thermal sciences wherein the systems are characterized by two major limitations. First, experimental measurements are limited to optical diagnostics in restricted regions of the flow (*e.g.*, a subsection of the flowfield using optical access along a single direction). Second, there is incomplete *a priori* specification of the boundary conditions for the flow simulation.

3 Description of Research

3.1 Objective

The objective of the research is the development of a general methodology for synergizing experiment and simulation for system evaluation in thermal sciences wherein the experimental measurements are restricted in region and scope, and the *a priori* boundary conditions for simulation are incomplete. The approach is firmly based on the DDDAS concept wherein the experiment directs the simulation and vice-versa in a closed-loop manner. The basic approach is to approximate the unknown boundary conditions by minimizing the error in the prediction

of the measured data (*i.e.*, the experiment driving the simulation), and to identify needed subsequent experimental measurements to reduce the error (*i.e.*, the simulation driving the experiment).

3.2 Configuration

We consider a rectangular jet injected perpendicular to an incompressible laminar boundary layer. The configuration is shown in Fig. 1. The inflow is an equilibrium laminar boundary layer in air defined by the specified freestream conditions (velocity U_∞ , static pressure p_∞ and static temperature T_∞) and boundary layer thickness¹ δ_∞ . The jet is defined by the jet average velocity U_j , static pressure p_j , and static temperature T_j .

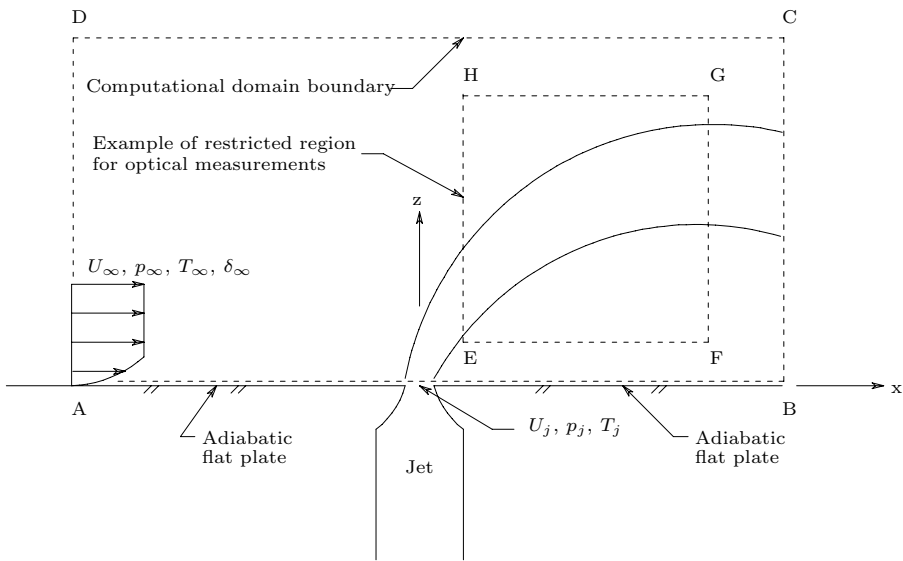


Fig. 1. Flow configuration

The computational domain ABCD is shown in Fig. 1. The inflow conditions (U_∞ , p_∞ , T_∞ and δ_∞) are assumed known. By analogy to the optical fiber furnace, the jet exit temperature T_j is assumed unknown, while the jet exit pressure p_j and average velocity U_j are assumed known. In future work, more complex problems will be considered. The specific values are indicated in Table 1.

The objective of this initial research problem is the determination of the jet exit temperature T_j through a DDDAS methodology using a closed-loop iteration of experiment and simulation. The experimental data are diode laser absorption measurements [8, 9, 10]. The simulation is performed using the commercial

¹ Here, δ_∞ is the laminar boundary layer thickness that would exist at the location of the jet exit in the absence of the jet. Equivalently, the distance from the leading edge of the flat plate to the jet exit may be specified.

Table 1. Flow Conditions

<i>Quantity</i>	<i>Value</i>
U_∞ (m/s)	5
p_∞ (kPa)	101
T_∞ (K)	298
U_j (m/s)	15
p_j (kPa)	101
T_j (K)	TBD
L (mm)	158
W (mm)	3.2

LEGEND

- L distance from leading edge to jet
- W width of jet
- TBD to be determined by DDDAS

Fluent software [11] which solves the incompressible Navier-Stokes equations using a spatially and temporally second-order accuracy finite volume algorithm.

3.3 DDDAS Methodology

A DDDAS methodology is used to determine the optimal location of the experimental measurements through an iterative procedure involving experiment and simulation. The methodology is illustrated in Fig. 2. Consider a diode laser absorbance measurement in a vertical plane at a specific location (x, z) within a limited region EFGH shown in Fig. 1. The relative instantaneous absorbance $\mathcal{A}(x, z, t)$ over a spanwise distance $y_1 \leq y \leq y_2$ is defined as

$$\mathcal{A}(x, z, t) = \frac{(I_o - I(x, z, t))}{I_o} \tag{1}$$

where I_o is the incident intensity at (x, y_1, z) and $I(x, z, t)$ is the instantaneous transmitted intensity at (x, y_2, z) . The absorbance $\mathcal{A}(x, z, t)$ can also be accurately calculated from Beer-Lambert’s Law

$$\mathcal{A}(x, z, t) = 1 - \exp \left[- \int_{y_1}^{y_2} \kappa_\nu dy \right] \tag{2}$$

assuming the diode laser frequency ν , mean partial pressure $p_i(x, y, z, t)$ of the specific absorbing species (*e.g.*, O_2) and static temperature $T(x, y, z, t)$ are known. The absorbance per cm of the $^Q R_2(6)$ line of the oxygen $b_1 \Sigma_g^+ \nu' = 0 \leftarrow X^3 \Sigma_g^-, \nu'' = 0$ transition at 761.139 nm is

$$\frac{d\mathcal{A}}{dy} = 0.083 T^{-1} - 2.26 \cdot 10^{-5} \tag{3}$$

where $T(x, y, z, t)$ is the static temperature in K. We identify the experimental measured absorbance in (1) by $\mathcal{A}_e(x, z, t)$ and the simulated absorbance in (2) by $\mathcal{A}_s(x, z, t)$.

Since the flowfield is unsteady as discussed below, the experimental and simulated absorbance power spectra $\mathcal{P}_e(f; x, z)$ and $\mathcal{P}_s(f; x, z)$ are compared where f is the frequency. At a given (x, z) -location, we may define the L_2 norm of the difference between the experimental and simulated absorbance power spectra as

$$\mathcal{E}(x, z) = \int_0^\infty [\mathcal{P}_e(f; x, z) - \mathcal{P}_s(f; x, z)]^2 df \quad (4)$$

The integral (4) is computed at the specific multiple (x, z) -locations of the experiment. An rms value of the error \mathcal{E}_{rms} is computed as

$$\mathcal{E}_{rms} = \frac{1}{n} \sum_i \mathcal{E}(x_i, z_i) \quad (5)$$

where n is the number of experimental location measurements.

The error \mathcal{E}_{rms} is a function of the assumed value for the jet exit temperature T_j . We determine a revised value for T_j by minimizing the error \mathcal{E}_{rms} using the Sequential Quadratic Programming procedure CFSQP [12] developed at the University of Maryland.

The flowfield is recomputed using the updated T_j and then the flowfield is searched to find the x -location where the Total Variation of $\partial \bar{\mathcal{A}}_s(x, z)/\partial z$ is the largest, where $\bar{\mathcal{A}}_s$ is the time-averaged absorbance at (x, z) and the Total Variation is defined by

$$\text{TV}(\bar{\mathcal{A}}_s(x)) = \int_{z_1}^{z_2} \left| \frac{\partial \bar{\mathcal{A}}_s}{\partial z} \right| dz \quad (6)$$

This heuristic algorithm is based on the concept that the optimal x -plane location for the next experimental measurement is characterized by the largest variations in the experimental measurement over the x -plane.

4 Results

We present the preliminary results of our DDDAS methodology (Fig. 2) for evaluation of the fluid-thermal system shown in Fig. 1 based upon the first four months of our research project. A time-accurate Navier-Stokes simulation was performed for the flow conditions shown in Table 1 using an initial jet temperature $T_j = 398$ K. Instantaneous temperature contours² are shown in Figs. 3 to 6 at intervals³ of 40 ms. The flowfield is strongly unsteady. The hot jet fluid is entrained into large spanwise vortices which convect downstream. Since the absorbance is inversely proportional to temperature (see (3)), the instantaneous absorbance signal at a fixed location (x, z) varies as each vortex is convected past.

² The axes are in meters.

³ For comparison, the time required for the freestream flow to traverse from the left to right boundaries of the computational domain is 155 ms.

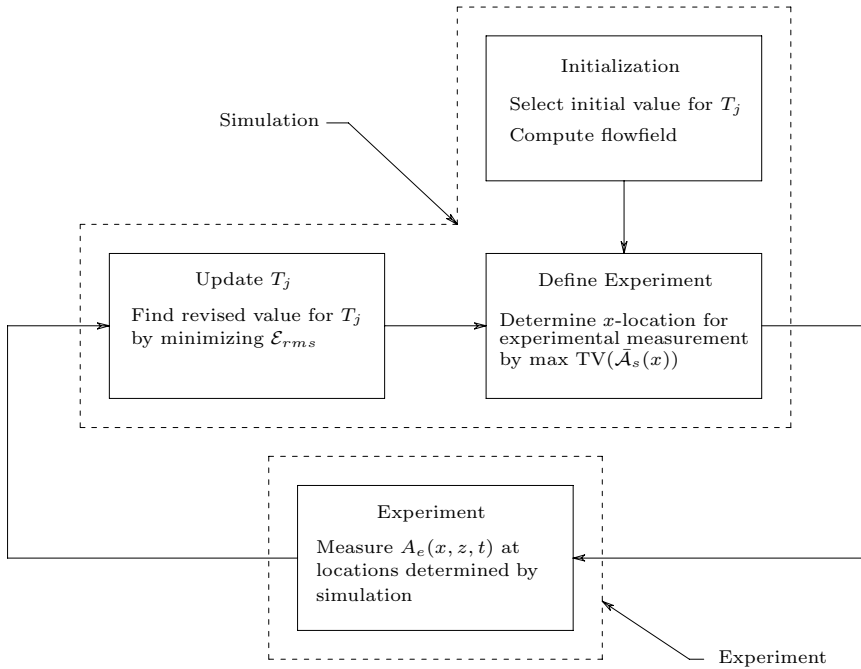


Fig. 2. DDDAS Methodology

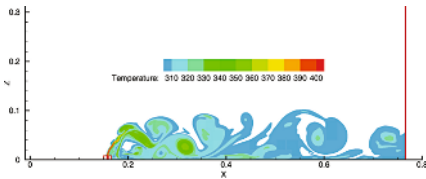


Fig. 3. $t = 0$

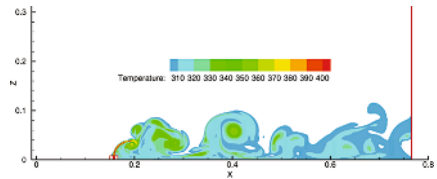


Fig. 4. $t = 40$ ms

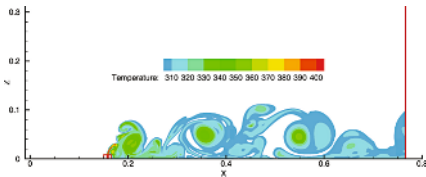


Fig. 5. $t = 80$ ms

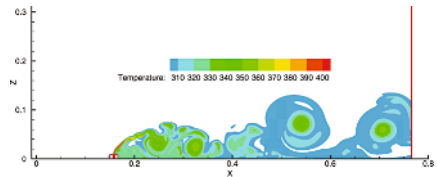


Fig. 6. $t = 120$ ms

An experimental facility (Fig. 7) has been constructed in the low speed wind tunnel at Rutgers University for the problem configuration (Fig. 1). Initial measurements of oxygen absorption were taken 30 mm downstream of the jet exit and 30 mm above the wall based upon the simulation results which indicated that the temperature fluctuations would be large at this location (see Figs. 3 to

6). The experimental inflow conditions were similar to the simulation; however, the jet velocity $U_j = 8.7$ m/s. The absorption path was a dual-pass geometry with the laser traversing the flowfield twice at a half angle of 0.2 degrees. Therefore, the effective spatial resolution of the diagnostic is 5 mm. The raw absorbance data traces were reduced to find temperature as a function of time using (3).

The temperature results display a highly unsteady behavior similar to the simulation. The power spectrum of the experimental temperature fluctuations (Fig. 8) shows evidence of vortex shedding behavior with characteristic frequencies at 8 and 15 Hz.

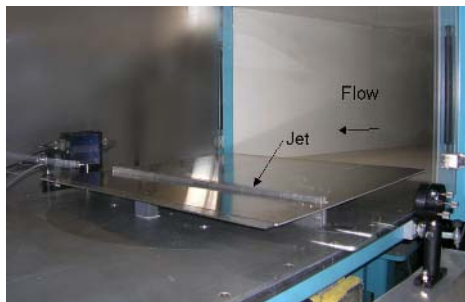


Fig. 7. Experimental configuration

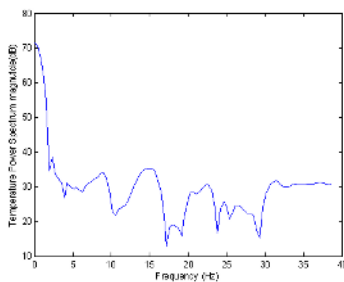


Fig. 8. \mathcal{P}_e at $x = 30$ mm, $z = 30$ mm

5 Future Work

Our immediate future work will focus on three problems. First, we will complete the determination of the jet temperature T_j (and therefore the complete evaluation of the fluid-thermal system in Fig. 1) using our DDDAS methodology. Second and third, we will extend the complexity of the problem to two *a priori* unknown boundary conditions (T_j and U_j) and then three *a priori* unknown boundary conditions (T_j , U_j and p_j). This sequence of problems will validate the accuracy and robustness of our DDDAS approach.

6 Conclusions

A methodology for evaluation of fluid-thermal systems is developed based upon the Dynamic Data Driven Application Systems approach. The methodology is intended for fluid-thermal systems where complete specification of the boundary conditions is not known *a priori* and experimental measurements are restricted to a subregion of the fluid-thermal domain. The methodology synergizes experiment and simulation in a closed-loop, iterative manner to achieve a full evaluation of the fluid-thermal system. Preliminary results are presented for the configuration of a heated jet injected into a laminar boundary layer where the jet temperature is not known *a priori*. Future work is described.

Acknowledgments

The research is sponsored by the US National Science Foundation under grant CNS-0539152 (1 Oct 2005 - 30 Sept 06). The program manager is Dr. Frederica Darema.

References

1. Y. Jaluria. *Design and Optimization of Thermal Systems*. McGraw-Hill, New York, 1998.
2. J. Issa. Temperature Measurement in an Optical Fiber Draw Furnace. MS Thesis, Dept of Mechanical and Aerospace Engineering, Rutgers University, 1995.
3. S. Roy Choudhury and Y. Jaluria. Practical Aspects in the Drawing of an Optical Fiber. *Journal of Materials Research*, 13:483–493, 1998.
4. Y. Jaluria. Thermal Processing of Materials: From Basic Research to Engineering. *Journal of Heat Transfer*, 125:957–979, 2003.
5. Dynamic Data Driven Applications Systems (DDDAS) Website. National Science Foundation, <http://www.nsf.gov/cise/cns/darema/dddas/index.jsp>.
6. F. Darema. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In *Fourth International Conference on Computational Science*, pages 662–669, Berlin, 2004. Springer-Verlag.
7. Dynamic Data Driven Applications Systems (DDDAS) Homepage. <http://www.dddas.org/>.
8. S. Chou, D. Baer, and R. Hanson. Diode-laser Absorption Measurements of CH₃Cl and CH₄ Near 1.65 μm ". *Applied Optics*, 36:3288–3293, 1997.
9. R. Mihalcea, D. Baer, and R. Hanson. Diode Laser Sensor for Measurements of CO, CO₂ and CH₄ in Combustion Flows. *Applied Optics*, 36:8745–8752, 1997.
10. D. Nelson, M. Zahniser, J. McManus, C. Kolb, and J. Jimenez. A Tunable Diode Laser System for the Remote Sensing of On-road Vehicle Emissions. *Applied Physics B*, 67:433–441, 1998.
11. Fluent Version 6. Fluent, Inc. <http://www.fluent.com>.
12. C. Lawrence, J. Zhou, and A. Tits. Users Guide for CFSQP Version 2.3: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints. Technical Report TR-94-16r1, Institute for Systems Research, University of Maryland, November 1994.

Inversion of Airborne Contaminants in a Regional Model

Volkan Akcelik¹, George Biros², Andrei Draganescu⁴, Omar Ghattas³, Judith Hill⁴,
and Bart van Bloemen Waanders⁴

¹ Stanford Linear Accelerator Center
volkan@slac.stanford.edu

² Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
biros@seas.upenn.edu

³ Institute for Computational Engineering and Sciences
The University of Texas at Austin
omar@ices.utexas.edu

⁴ Optimization and Uncertainty Estimation Department
Sandia National Laboratories
{aidraga, jhill, bartv}@sandia.gov

Abstract. We are interested in a DDDAS problem of localization of airborne contaminant releases in regional atmospheric transport models from sparse observations. Given measurements of the contaminant over an observation window at a small number of points in space, and a velocity field as predicted for example by a mesoscopic weather model, we seek an estimate of the state of the contaminant at the beginning of the observation interval that minimizes the least squares misfit between measured and predicted contaminant field, subject to the convection-diffusion equation for the contaminant. Once the “initial” conditions are estimated by solution of the inverse problem, we issue predictions of the evolution of the contaminant, the observation window is advanced in time, and the process repeated to issue a new prediction, in the style of 4D-Var. We design an appropriate numerical strategy that exploits the spectral structure of the inverse operator, and leads to efficient and accurate resolution of the inverse problem. Numerical experiments verify that high resolution inversion can be carried out rapidly for a well-resolved terrain model of the greater Los Angeles area.

1 Introduction

We are interested in a DDDAS problem of localization of airborne contaminant releases in regional atmospheric transport models from sparse observations. Given measurements of the contaminant over an observation window at a small number of points in space, and a velocity field as predicted for example by a mesoscopic weather model, we seek an estimate of the state of the contaminant at the beginning of the observation interval that minimizes the least squares misfit between measured and predicted contaminant field, subject to the convection-diffusion equation for the contaminant. Once the “initial” conditions are estimated by solution of the inverse problem, we issue predictions of the evolution of the contaminant, the observation window is advanced in time, and the process repeated to issue a new prediction, in the style of 4D-Var.

The main difficulty of this inverse problem is that the inverse operator is formally dense and of grid dimension, and requires as many forward/adjoint solves to create as the number of grid points. Thus, the inverse problem is intractable for highly-resolved problems, even with the most powerful supercomputers available today. To address this issue, we invoke a conjugate gradient method that is tailored to the spectral structure of the inverse operator, in particular that the operator behaves like a Fredholm integral equation of the second kind. Thus, convergence can be obtained in a mesh-independent number of iterations, typically very small. Numerical experiments suggest that high resolution inversion can be carried out rapidly for a sufficiently well-resolved terrain model of the greater Los Angeles area.

In Section 2 we formulate the inverse problem as an output least squares optimization problem with a convection-diffusion PDE constraint. First order optimality conditions produce a coupled system of partial differential-algebraic equations, which includes the initial value convection-diffusion PDE, the terminal-value adjoint convection-diffusion PDE, and an algebraic equation for the initial concentration. This system is an ill-posed boundary value problem in 4D space-time; block-elimination to a 3D system in just the initial condition variables creates the aforementioned dense inverse operator (a Schur complement) that is of dimension of the grid points. A conjugate gradient method then makes effective use of its spectral structure.

Section 3 provides a prototype inversion scenario: localization of the release of a contaminant in the Los Angeles harbor from short-term measurements of its transport by onshore winds, followed by longer-term prediction of the transport of the contaminant throughout the Greater LA Basin. We conduct numerical experiments that demonstrate the capability and effectiveness of the inversion.

2 Formulation and Optimality Conditions

Mathematically, transport of the contaminant is described by the convection-diffusion equation. We seek to reconstruct the initial concentration from measurements of the concentration over a short time horizon, taken at a small number of sensor locations throughout the domain. In this section we give details on the mathematical formulation of the inverse problem, its discretization, and the strategy for its numerical solution.

Given observations of the concentration $\{u_j^*\}_{j=1}^{N_s}$ at N_s locations $\{x_j\}_{j=1}^{N_s}$ inside a domain Ω , we wish to estimate the initial concentration $u_0(\mathbf{x})$ that leads to the closest reproduction of the observed concentrations. The inverse problem is formulated as a constrained, regularized, least squares optimization problem:

$$\min_{u, u_0} \mathcal{J}(u, u_0) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{j=1}^{N_s} \int_0^T \int_{\Omega} (u - u_j^*)^2 \delta(\mathbf{x} - \mathbf{x}_j) \, d\mathbf{x} \, dt + \frac{\beta}{2} \int_{\Omega} u_0^2 \, d\mathbf{x},$$

$$\text{subject to} \quad \frac{\partial u}{\partial t} - \nu \Delta u + \mathbf{v} \cdot \nabla u = 0, \quad \text{in } \Omega \times (0, T), \tag{1}$$

$$\nu \nabla u \cdot \mathbf{n} = 0, \quad \text{on } \Gamma \times (0, T),$$

$$u = u_0, \quad \text{in } \Omega \times \{t = 0\}.$$

The first term in the objective functional \mathcal{J} represents a least-squares misfit of predicted concentrations $u(\mathbf{x}_j)$ with observed concentrations $u^*(\mathbf{x}_j)$, where the delta function localizes the u and u^* fields to the sensor locations. The second term in \mathcal{J} , scaled by the constant $\beta/2$, is a regularization term that introduces well-posedness in the inverse problem. In the absence of the regularization term, the inverse problem would be ill-posed even if measurements are available at all points in space and time: small perturbations in measurements, e.g. due to sensor faultiness or data transmission, would produce exponentially large errors in the recovered solution. It is also the case that we cannot hope to recover components of the initial concentration that are much more oscillatory than dictated by the spacing of the sensors. Therefore, oscillatory components of u_0 lie in the null space of the inverse operator and, in the absence of regularization, will appear as arbitrary noise in the reconstructed initial concentration field.

The constraints in the optimization problem (1) are the contaminant transport convection-diffusion equation, boundary condition, and initial condition, where $\mathbf{v}(\mathbf{x}, t)$ is the velocity field and ν is the diffusion coefficient. The transport of the pollutant is driven by the initial conditions, the diffusion, and the velocity field. In practice, the velocity field would be provided by a regional numerical weather prediction model. For our present purposes, however, we are interested in assessing the viability of our inversion method, and thus for simplicity, we employ a steady laminar incompressible Navier-Stokes solver to generate wind velocity fields over a terrain of interest.

The inverse problem then is to determine the initial concentration field $u_0(\mathbf{x})$, and the resulting space-time evolution of the concentration $u(\mathbf{x}, t)$, by solving the optimization problem (1). First-order necessary conditions for optimality, the *KKT conditions*, may be derived by introducing a Lagrangian functional:

$$\begin{aligned} \mathcal{L}(u, u_0, p) \stackrel{\text{def}}{=} & \frac{1}{2} \sum_{j=1}^{N_s} \int_0^T \int_{\Omega} (u - u^*)^2 \delta(\mathbf{x} - \mathbf{x}_j) \, d\mathbf{x} \, dt + \frac{\beta}{2} \int_{\Omega} u_0^2 \, d\mathbf{x} \\ & + \int_0^T \int_{\Omega} \left(p \frac{\partial u}{\partial t} + \nu \nabla u \cdot \nabla p + p \mathbf{v} \cdot \nabla u \right) \, d\mathbf{x} \, dt + \int_{\Omega} p(u - u_0) \, d\mathbf{x}, \end{aligned} \tag{2}$$

in which the *adjoint concentration* $p(\mathbf{x}, t)$ is used to enforce the convection-diffusion equation and initial condition. Requiring stationarity of the Lagrangian \mathcal{L} with respect to p , u , and u_0 , respectively yields the KKT conditions, which consist of:

The forward convection-diffusion problem

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u + \mathbf{v} \cdot \nabla u &= 0 \quad \text{in } \Omega \times (0, T), \\ \nu \nabla u \cdot \mathbf{n} &= 0 \quad \text{on } \Gamma \times (0, T), \\ u &= u_0 \quad \text{in } \Omega \times \{t = 0\}. \end{aligned} \tag{3}$$

The adjoint convection-diffusion problem

$$-\frac{\partial p}{\partial t} - \nu \Delta p - \nabla \cdot (p\mathbf{v}) = - \sum_{j=1}^{N_s} (u - u^*) \delta(\mathbf{x} - \mathbf{x}_j), \quad \text{in } \Omega \times (0, T)$$

$$\begin{aligned}
 (\nu \nabla p + vp) \cdot \mathbf{n} &= 0, \quad \text{on } \Gamma \times (0, T), \\
 p &= 0 \quad \text{in } \Omega \times \{t = T\}.
 \end{aligned}
 \tag{4}$$

The initial concentration equation

$$\beta u_0 - p|_{t=0} = 0 \quad \text{in } \Omega.
 \tag{5}$$

The equations in (3) are the original forward convection-diffusion transport problem for the contaminant field. The adjoint convection-diffusion problem (4) resembles the forward problem, but with some essential differences. First, it is a terminal value problem, the adjoint p is specified at the final time $t = T$, rather than an initial value problem. Second, the convection is directed backward along the streamlines. Finally, it is driven by a source term given by the negative of the misfit between the predicted and measured concentrations at sensor locations. The initial concentration equation (5) is, in the case of L^2 regularization, an algebraic equation. Together, (3), (4), and (5) furnish a coupled system of linear PDEs for (u, p, u_0) . The principal difficulty in solving this system is that, though the forward and adjoint transport problems are parabolic-hyperbolic problems, the KKT optimality system is a *coupled boundary value problem in 4D space-time*.

To simplify discussion of solution approaches, we introduce the operators A, T, B and R . Here, A denotes the forward transport operator and A^{-1} its inverse; T extends a spatial field at initial time into space-time; B is an observation operator that localizes space-time to points at which sensors are placed; R is the regularization operator (in the present case the identity); A^* is the adjoint transport operator and A^{-*} its inverse; and T^* restricts a space-time field to a spatial field at $t = 0$. With these definitions, we can write the KKT conditions in operator form:

$$\begin{bmatrix} B & 0 & A^* \\ 0 & \beta R & -T^* \\ A & -T & 0 \end{bmatrix} \begin{bmatrix} u \\ u_0 \\ p \end{bmatrix} = \begin{bmatrix} Bu^* \\ 0 \\ 0 \end{bmatrix}
 \tag{6}$$

Special-purpose Krylov solvers and parallel preconditioners can be very effective at solving discretized versions of optimality systems such as (6) for optimization problems constrained by *steady-state* PDEs [1, 2]. Here, however, the 4D space-time nature of (6) presents prohibitive memory requirements for large scale problems. Solution of (6) in its *full-space* form is essentially intractable for such problems using present computing resources. Instead, we pursue a *reduced-space* method that amounts to a block elimination combined with a matrix-free Schur complement solver.

Eliminating the concentration u and forward transport equation (third row of (6)) and adjoint concentration p and adjoint transport equation (first row of (6)) from the KKT optimality system, we obtain the Schur complement system for the initial concentration u_0 :

$$Hu_0 = g,
 \tag{7}$$

where the *reduced Hessian* (or inverse) operator H is defined by

$$H \stackrel{\text{def}}{=} T^* A^{-*} B A^{-1} T + \beta R,
 \tag{8}$$

and the *reduced gradient* g is defined by

$$g \stackrel{\text{def}}{=} T^* A^{-*} B u^*.$$

It is immediately clear that H is a symmetric and boundedly invertible operator. Thus, the optimization problem has a unique solution for non-vanishing β .

Notice that H is a non-local operator and while explicit formation of H and its singular value decomposition constitutes an attractive and popular approach for small-scale inverse problems [3], alternative approaches are essential for large-scale problems, and in particular those for which rapid response is mandated. Therefore, we opt to solve the discretized form of (7) using the conjugate gradient (CG) method. H is never formed explicitly; instead, we compute $w = Hv$, the action of the reduced Hessian on a given spatial field v , in matrix-free fashion as follows. (i) Set $u_0 = v$ and solve the forward transport equation (3) to obtain the concentration evolution u . (ii) Compute the misfit between the measurements u^* and the predicted concentrations u at the sensor locations. (iii) Use this misfit as a source to solve the adjoint transport equation (4) backward in time to obtain $p|_{t=0}$, the adjoint at $t = 0$. (iv) Set $w = \beta v - p|_{t=0}$. Therefore, each application of the reduced Hessian requires solving two transport equation, one forward in time and one backward. Besides u_0 , we need to store the entire time history of u (if we have measurements over the entire time interval $(0, T)$) but *only* at the sensor locations. In contrast with full-space methods, we avoid storing the forward and adjoint concentration time histories. Thus, the memory requirements for the inverse problem (1) are similar to those of the forward problem.

The overall computational cost of the (unpreconditioned) CG method is the work per iteration — dominated by the two transport equations solutions — multiplied by the number of CG iterations. The latter depends on the condition number of the reduced Hessian. One can show that, for fixed β , H is a compact perturbation of the identity, and its discrete version, denoted by H_h (h represents the discretization parameter), has a mesh-independent condition number [4]. Furthermore, its spectrum has a small number of clusters; it collapses exponentially onto β . Therefore, if we use a Krylov method, such as CG, to solve (7), the number of iterations for a specific relative residual reduction will also be mesh-independent. This is an optimal number of iterations, and we have verified numerically [5]. Thus, mesh-independent convergence comes for free, with the constant mildly deteriorating with $\beta \rightarrow 0$. The constant also depends on the Peclet number, the length of the time horizon, the complexity of the velocity field, and the topography.

3 Implementation and Numerical Results for Los Angeles Region

We demonstrate our inversion framework on a hypothetical atmospheric contamination event in the Greater Los Angeles Basin (GLAB) region. Using real topographical data and synthesized velocity fields, we conduct numerical experiments in which sparse observations are extracted from forward simulations and subsequently used in the inverse problem. Our implementation builds on PETSc [6] to manage parallel data structures, interface with linear solvers and domain decomposition preconditioners, and utilize a range of software services.

In our actual implementation, we first discretize the optimization problem (1), and then write optimality conditions (as opposed to the writing the infinite-dimensional optimality conditions (6) and then discretizing; in the present case the two are not identical [7]). We employ Streamline Upwind Petrov-Galerkin (SUPG) finite elements [8] in space and a Crank-Nicolson discretization in time. For problems with high Peclet number, stabilized methods such as SUPG are more accurate than standard Galerkin on coarse meshes. We use a logically-rectangular topography-conforming isoparametric hexahedral finite element mesh on which piecewise-trilinear basis functions are defined. Since the Crank-Nicolson method is implicit, we “invert” the time-stepping operator using a restarted GMRES method, accelerated by an additive Schwarz domain decomposition preconditioner, both from the PETSc library.

Contaminant transport is modeled over a $360 \text{ km} \times 120 \text{ km} \times 5 \text{ km}$ region in the GLAB. Land surface elevations are obtained at 1 km spacing from the USGS Land Processes Distributed Active Archive Center (GTOPO30 digital elevation model).¹ The three-dimensional mesh is created from the surface elevations by inserting equally spaced grid points vertically from the surface grid to the top of the domain at 5 km.

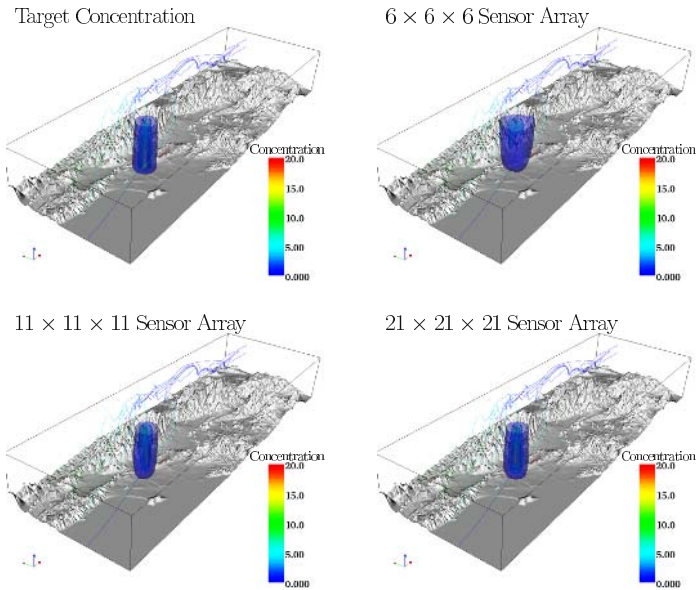


Fig. 1. Sensitivity of the inversion result to the sensor array density. The target initial concentration is shown in the upper-left corner, and inversion results using successively-finer sensor arrays are shown in the subsequent images. As the number of sensors in each direction increases, the quality of the reconstruction of the initial concentration plume improves. Inversion using the $21 \times 21 \times 21$ sensor array takes 2.5 hours on 64 processors of the Alphaserver EV68 system at the Pittsburgh Supercomputing Center.

¹ http://edcdaac.usgs.gov/gtopo30/dem_img.asp

To simulate a contamination event, an initial contaminant plume with a Gaussian concentration given by $20 \exp(-0.04|x - x_c|)$ is centered at $x_c = (120 \text{ km}, 60 \text{ km}, 0 \text{ km})$. The plume is transported over the GLAB region by solving the forward convection-diffusion equation with specified velocity field over a time horizon of 120 minutes. Sensor measurements are taken every 3 minutes to develop a time history from which to invert. For this contaminant, the diffusion coefficient is taken as $\nu = 0.05$. The regularization parameter is fixed at $\beta = 0.01$. The forward and inverse problems are solved on a mesh with $361 \times 121 \times 21$ grid points, representing 917,301 concentration unknowns at each time step. The time step is the same as the sensor recording rate, i.e. 3 minutes, for a total of 40 time steps. Therefore, there are about 74×10^6 total space-time variables in the KKT optimality system (6).

A velocity field \mathbf{v} for the convection-diffusion equation is synthesized by solving the steady-state incompressible Navier-Stokes equations, $\rho(\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p - \mu\Delta\mathbf{v} = \mathbf{0}$, $\nabla \cdot \mathbf{v} = 0$, where p is the fluid pressure and (ρ, μ) are its density and viscosity. To simulate an onshore wind, an inflow Dirichlet boundary condition with $v_x = v_{\max}(z/(5.0 - z_{\text{surface}}))^{0.1}$ and zero for the other components is applied to the $x = 0$ plane, where v_{\max} is specified as 30 km/hr. Traction-free boundary conditions are applied to the outflow plane at $x = 360 \text{ km}$. Traction-free tangential and no-slip normal boundary conditions are applied to the remaining portions of the boundary. An SUPG-stabilized finite element method is employed to solve the Navier-Stokes equations using linear tetrahedral elements derived by subdividing the convection-diffusion hexahedral mesh.

We sample the concentrations from the forward transport simulations on a uniformly-spaced array of sensors, and use them as synthetic observations to drive the inverse problem. Figure 1 depicts inversion results for different sensor array densities, along with the actual initial concentration (labeled *Target* in the figure). One of the critical issues is to determine the number of sensors required to resolve the initial concentration. Recall that due to the non-vanishing regularization parameter and the fixed mesh size, we cannot expect to recover the initial concentration exactly.

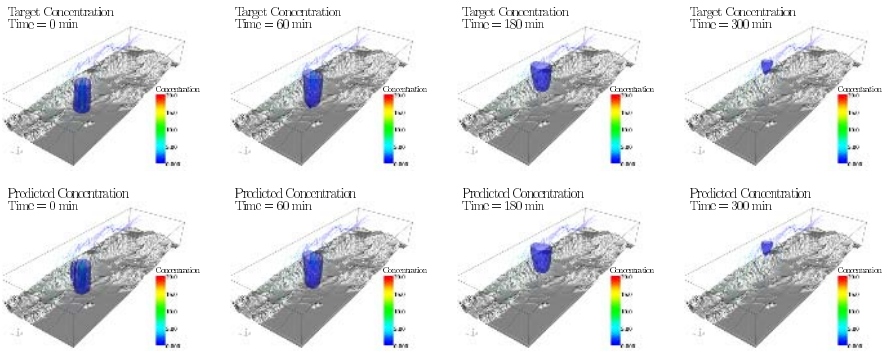


Fig. 2. Illustration of the predictive capabilities of our inversion algorithm, using an $11 \times 11 \times 11$ sensor array. Forward transport of the actual initial concentration is compared with forward transport of the reconstructed initial concentration plume. The trajectories are close to each other, and the comparison improves with time.

What is of ultimate interest is how successful the reconstructed initial field is in predicting the actual transport of the contaminant. Figure 2 compares the actual evolution (left) and predicted evolution (right) of the contaminant plume in time. It is evident from this figure that although the reconstructed concentration does not match the actual concentration exactly at $t = 0$, the difference between the two diminishes over time, due to the dissipative nature of the forward convection-diffusion problem.

Acknowledgements

Supported in part by the Computer Science Research Institute at Sandia National Laboratories, the U.S. Department of Energy under the SciDAC Terascale Optimal PDE Simulations (TOPS) Center and grant DE-FG02-04ER25646, and the National Science Foundation under grants ACI-0121667, EAR-0326449, CCF-0427985, and CNS-0540372. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000. Computing resources at the Pittsburgh Supercomputing Center provided under NSF TeraGrid award MCA04N026P.

References

1. Biros, G., Ghattas, O.: Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM Journal on Scientific Computing* **27**(2) (2005) 687–713
2. Biros, G., Ghattas, O.: Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange Newton solver, and its application to optimal control of steady viscous flows. *SIAM Journal on Scientific Computing* **27**(2) (2005) 714–739
3. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM (1997)
4. Drăgănescu, A.: Two investigations in numerical analysis: Monotonicity preserving finite element methods, and multigrid methods for inverse parabolic problems. PhD thesis, University of Chicago (2004)
5. Drăgănescu, A., Dupont, T.F.: Optimal order multi-level preconditioners for regularized ill-posed problems (2005) To appear.
6. Balay, S., Buschelman, K., Gropp, W.D., Kaushik, D., McInnes, L.C., Smith, B.F.: PETSc home page. <http://www.mcs.anl.gov/petsc> (2001)
7. Abraham, F., Behr, M., Heinkenschloss, M.: The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations. *Finite Elements in Analysis and Design* **41**(3) (2004) 229–251
8. Brooks, A.N., Hughes, T.J.R.: Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* **32** (1982) 199–259

Data Assimilation Using the Global Ionosphere-Thermosphere Model*

I.S. Kim¹, J. Chandrasekar¹, A. Ridley², and D.S. Bernstein¹

¹ Department of Aerospace Engineering, University of Michigan, Ann Arbor
dsbaero@umich.edu

² Department of Atmospheric, Oceanic and Space Sciences,
University of Michigan, Ann Arbor

Abstract. We consider a data assimilation technique for coupled ionospheric and thermospheric dynamics. The Global Ionosphere-Thermosphere Model (GITM) is used to simulate the ionospheric and thermospheric dynamics, and evaluate the performance of the data assimilation scheme that estimates the ion densities and flow speeds. This estimation technique is based on the state dependent Riccati equation (SDRE), which uses a frozen linear dynamics matrix for the time update of the error covariance and the evaluation of the Kalman filter gain. We demonstrate the performance of the data assimilation technique on a section of the ionosphere.

1 Data Assimilation for Space Weather Prediction

The Sun drives our atmosphere in many ways. The most commonly understood way is through the heating of the lower atmosphere where sunlight heats the atmosphere causing weather. A similar process occurs in the thermosphere (above about 60 miles), where the thin atmosphere absorbs sunlight, and the atmosphere is heated dramatically. Another way in which the Sun affects the upper atmosphere is through the transfer of electromagnetic energy. The Sun's atmosphere flows away from it at supersonic speeds and encounters the Earth's magnetic field, transferring a significant amount of electromagnetic energy to our magnetosphere. This energy flows throughout the magnetosphere until it either leaves the system or is deposited into the upper atmosphere. This energy takes the form of strong electromagnetic currents and aurora, that is, the Northern and Southern lights.

During quiet times, when the Sun's atmosphere is calm, the heating of the atmosphere through sunlight is the dominant process. But the Sun sometimes erupts, sending large amounts of plasma and strong magnetic fields toward the Earth in events called coronal mass ejections (CMEs). When CMEs reach the Earth, they cause the Earth's magnetosphere to shrink significantly, bringing the aurora to much lower latitudes (sometimes over the United States). During these

* This research was supported by the National Science Foundation under grants CNS-0539053 and ATM-0325332.

disturbed periods, the electromagnetic energy injected into the atmosphere can become the dominant process, driving the atmosphere in ways that we do not fully comprehend.

In 1989, the entire province of Quebec, Canada, experienced a power blackout for 9 hours as a result of a CME. More recently, in October 2003, the so-called “Halloween storms” caused major distortions of the Earth’s radiation belt and magnetic field, disrupting radio communication systems. These types of solar storms are known or suspected to have damaged several satellites in orbit around the Earth, in particular, an AT&T Telestar 401 satellite in 1997, a PanAmSat Galaxy IV satellite in 1998, and a Kodama communications satellite in 2003 (as a result of the Halloween storms). One of the primary mechanisms for damage, especially in the electric grid, is fluctuations in the magnetic field, which induce currents through Faraday’s law.

Solar storms also degrade the accuracy of the Global Positioning System, which is used for military and civilian positioning systems. This degradation occurs because the GPS signals travel through the ionosphere, which modifies their travel time from the satellite to the receiver. If there are strong changes in the ion density in the ionosphere, then the normal correction for the ionosphere is no longer accurate, causing errors in position. While this navigation error might not matter to the typical automobile, it has significant ramifications for aircraft during landings in poor visibility as well as for GPS-guided missiles.

Modeling and prediction of the upper atmosphere and the near-Earth space environment is generally referred to as *space weather*. This relatively new field is challenging because of the limited number of measurements in space. However, the situation has started to improve with increasing awareness of the serious effects that space weather has on terrestrial and space-based systems.

The present paper describes a Kalman filter for data assimilation based on a coupled model of the thermosphere and ionosphere. For this purpose we use the GITM model [5], which contains various species of both ions and neutrals, as well as temperatures and winds for both ions and neutrals. The domain of the model extends from 100 km to approximately 600 km altitude over the entire globe in latitude and longitude. We present preliminary data assimilation results based on a one-dimensional (vertical) model of the atmosphere. Future work will involve extensions to 3-dimensional data assimilation, assimilation of data from incoherent scatter radar, and techniques for using the data assimilation results to redirect the radar during real-time operation, that is, targeted data assimilation.

2 Kalman Filtering for GITM

Classical Kalman filtering is based on the linear dynamics and measurements

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + w_k \\ y_k &= C_k x_k + v_k.\end{aligned}$$

The objective is to use the measurements y_k to estimate unmeasured state components. The *forecast step* consists of the simulation and measurement step

$$x_{k+1}^f = A_k x_k^{\text{da}} + B_k u_k, \quad y_k^f = C_k x_k^f,$$

while the *data assimilation step* is given by the “closed-loop” data-driven update

$$x_k^{\text{da}} = x_k^f + K_k (y_k - y_k^f).$$

Here, x_k^f and x_k^{da} are the *forecast estimate* and the *data assimilation estimate*. The data-injection gain K_k involves the error covariance matrices updated by

$$P_k^{\text{da}} = (I - K_k C_k) P_k^f, \quad P_{k+1}^f = A_k P_k^{\text{da}} A_k^T + Q_k, \\ K_k = P_k^f C_k^T (R_k + C_k P_k^f C_k^T)^{-1}.$$

Here, Q_k and R_k are the covariances of w_k and v_k , respectively.

Since GITM is nonlinear, nonlinear estimation algorithms are needed. One approach is the extended Kalman filter (XKF) in which the nonlinear dynamics advance the estimate, while the Jacobian of the dynamics propagates the covariance. Alternatively, the ensemble Kalman filter (EnKF) [4, 3] is often used. The size of the ensemble needed to obtain accurate estimates is crucial [4].

An alternative approach is the “frozen-linear” state-dependent Riccati equation approach, wherein the nonlinear dynamics advance the estimate, while an exact, state-dependent dynamics matrix propagates the pseudo covariance [2]. Hence, we rewrite the nonlinear dynamics

$$x_{k+1} = f(x_k, k) + B(x_k) u_k$$

in the factored form

$$x_{k+1} = A(x_k, k) x_k + B(x_k) u_k.$$

This factorization is not unique, and there do not exist general principles for constructing favorable factorizations [1]. However, since no Jacobian exists for finite volume schemes with nondifferentiable limiters, the use of an SDRE factorization seems appropriate. Stability is discussed in [2]. For the system considered in the present paper, we use an SDRE factorization whose performance is similar to the performance of an extended Kalman filter in which the points of discontinuity are ignored.

3 Model Description

GITM models the thermosphere and ionosphere using a stretched 3D spherical grid in latitude and altitude. The number of grid points can be specified, so that the resolution is extremely flexible. GITM solves for the neutral densities of O, O₂, N(²D), N(²P), N(⁴S), N₂, NO, H, and He; and ion species O⁺(⁴S), O⁺(²D), O⁺(²P), O₂⁺, N⁺, N₂⁺, NO⁺, H⁺, and He⁺. By not assuming hydrostatic equilibrium, the vertical solver is different from other thermospheric codes. Furthermore, GITM can simulate above a particular location on the earth (one-dimensional) or a spherical section around the earth (three-dimensional). Here, we simulate a one-dimensional grid, and ignore horizontal transport but retain

vertical advection. Three-dimensional source terms are included, which includes processes such as ion drag.

3.1 Neutral Dynamics

In the thermosphere, the neutrals are treated as having individual vertical velocities \mathbf{u}_s , where s denotes the species. The mass density is the sum of the species densities

$$\rho = \sum_s M_s N_s, \tag{1}$$

where M_s is the molecular mass and N_s is the number density of species s . In the vertical (or radial) direction, the natural logarithm of the total mass density ρ and number densities N_s are used as the primitive variables

$$\mathcal{R} \triangleq \ln(\rho), \quad \mathcal{N}_s \triangleq \ln(N_s) \tag{2}$$

and the continuity equation for each neutral species is

$$\frac{\partial \mathcal{N}_s}{\partial t} + \nabla \cdot \mathbf{u}_s + \mathbf{u}_s \cdot \nabla \mathcal{N}_s = S_N, \tag{3}$$

where the source S_N includes eddy diffusion and chemical sources and losses.

Next, we define the normalized temperature

$$\mathcal{T} \triangleq p/\rho, \tag{4}$$

with p the total neutral pressure. The momentum equations for species s are

$$\frac{\partial \mathbf{u}_s}{\partial t} + \mathbf{u}_s \cdot \nabla \mathbf{u}_s + \frac{k}{M_s} \nabla T + \frac{k}{M_s} T \nabla \mathcal{N}_s = S_u, \tag{5}$$

where $T \triangleq \frac{\bar{m}_n}{k} \mathcal{T}$, \bar{m}_n is the number density weighted average mass, and the source term S_u includes friction between species, gravitational, centrifugal, and Coriolis forces. For numerical accuracy, the energy equation is expressed as the normalized temperature \mathcal{T} instead of the exponentially varying p as in

$$\frac{\partial \mathcal{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathcal{T} + (\gamma - 1) \mathcal{T} \nabla \cdot \mathbf{u} = S_T, \tag{6}$$

where

$$\mathbf{u} \triangleq \frac{1}{\rho} \sum_s M_s N_s \mathbf{u}_s, \tag{7}$$

$\gamma = 5/3$ is the ratio of specific heats and the source term S_T includes solar extreme ultraviolet heating and cooling terms as described in [6].

3.2 Ion Advection

Ion continuity equation in the vertical direction is given by

$$\frac{\partial \mathcal{N}_j}{\partial t} + v_r \frac{\partial \mathcal{N}_j}{r} = S_j, \tag{8}$$

where \mathcal{N}_j is the log of the number density of the ion species j and v_r is ion velocity. The source S_j is obtained from the chemistry between ions and neutrals.

We specify ions for advection using (8), while other ion densities are controlled by reactions [5] set by $\frac{\partial \mathcal{N}_j}{r} = 0$. O^+ is the dominant ion in the low collision region of the ionosphere, and the most important species to advect [7].

4 Numerical Schemes

We describe the numerical scheme used in GITM for solving the advection equations and obtain a parametrization $A(x_k, k)$ for SDRE-based data assimilation.

4.1 Spatial Discretization for Advection

We perform data assimilation on the ion species, which has greater number density variation than the neutral species. That is, although we advect both the neutral and ion species, we use measurements to directly update the estimates of only the ion species. We describe the numerical scheme used to advect the ion species as in [5]. Note (8) can be expressed as

$$\frac{\partial U}{\partial t} + T \frac{\partial}{\partial r} U = S, \tag{9}$$

where $U \in \mathbb{R}^{10}$ is defined by

$$U \triangleq [\mathcal{N}_{O^+(4S)} \mathcal{N}_{O_2^+} \mathcal{N}_{N_2^+} \mathcal{N}_{N^+} \mathcal{N}_{NO^+} \mathcal{N}_{O^+(2D)} \mathcal{N}_{O^+(2P)} \mathcal{N}_{H^+} \mathcal{N}_{He^+} v_r]^T, \tag{10}$$

$T \in \mathbb{R}^{10 \times 10}$ is defined by

$$T \triangleq \text{diag}(v_r, 0, \dots, 0), \tag{11}$$

and for $i = 1, \dots, 10$, the i th entry of $S \in \mathbb{R}^{10}$ denotes the source for the i th ion species. We advect only $O^+(4S)$; other states in U are updated using only source terms. It follows from [5] that the time update is given by

$$U_j(k+1) = U_j(k) - \frac{t_s(k)}{\Delta r_j} \left[T_j(k) \left(U_{j+\frac{1}{2}}(k) - U_{j-\frac{1}{2}}(k) \right) - \left(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}} \right) \right] + S_j(k) \tag{12}$$

where U_j denotes U at the center of the cell j and $U_{j+\frac{1}{2}}$ denotes U at the edge between cells j and $j + 1$. For all $j = 3, \dots, n - 2$, we evaluate $U_{j+\frac{1}{2}}$ by using the modified monotized central limiter [5]. Also, $F_{j+\frac{1}{2}}$ is the flux at the edge between cells j and $j + 1$ evaluated using the Lax-Friedrichs 2nd-order scheme.

The boundary conditions for the one-dimensional GITM model are $U_1(k)$, $U_2(k)$, $U_{n-1}(k)$, and $U_n(k)$. Hence, it follows from (12) that the dynamics are

$$x_{k+1} = f(x_k, u_k), \tag{13}$$

where the state vector $x_k \in \mathbb{R}^{10(n-4)}$ is defined by

$$x_k \triangleq [U_3(k)^T \dots U_{n-2}(k)^T]^T \tag{14}$$

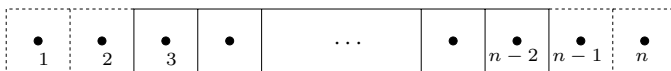


Fig. 1. One-dimensional grid used in the finite volume scheme

and $u_k \in \mathbb{R}^{10(4)+10(n-4)}$ is defined by

$$u_k \triangleq [U_1(k)^T U_2(k)^T U_{n-1}(k)^T U_n(k)^T S_3(k)^T \cdots S_{n-2}(k)^T]^T. \tag{15}$$

4.2 State-Dependent Frozen-Linear Representation

To express the update equation (13) as a state-dependent frozen-linear discrete-time equation, we write (12) as

$$U_j(k+1) = \mathcal{A}_{L2}^{[j]} U_{j-2}(k) + \mathcal{A}_{L1}^{[j]} U_{j-1}(k) + \mathcal{A}_M^{[j]} U_j(k) + \mathcal{A}_{R1}^{[j]} U_{j+1}(k) + \mathcal{A}_{R2}^{[j]} U_{j+2}(k) + S_j(k)$$

where $\mathcal{A}_{L2}^{[j]}, \mathcal{A}_{L1}^{[j]}, \mathcal{A}_M^{[j]}, \mathcal{A}_{R1}^{[j]}, \mathcal{A}_{R2}^{[j]} \in \mathbb{R}^{10 \times 10}$ depend on $U_i(k), i = j - 2, \dots, j + 2$, e.g.,

$$\mathcal{A}_{j,L2} = -\frac{t_s(k)}{\Delta r_j} \left\{ -\frac{1}{4} T_j(k) K_{j-1}^L - \frac{1}{4} \max(c_{j-\frac{3}{2}}, c_{j-\frac{1}{2}}) K_{j-1}^L \right\},$$

where K_j^L depends on the limiter [1] and $c_{j+\frac{1}{2}}$ is the maximum wave speed at the right edge of the j th cell [5].

Hence, it follows from (12) and (16) that

$$x_{k+1} = \mathcal{A}(x_k) x_k + \tilde{u}_k, \tag{16}$$

where \tilde{u}_k represents source terms and boundary conditions, and the block pentadiagonal matrix $\mathcal{A} \in \mathbb{R}^{10(n-4) \times 10(n-4)}$ is defined by

$$\mathcal{A}(x_k) \triangleq \begin{bmatrix} \mathcal{A}_M^{[3]} & \mathcal{A}_{R1}^{[3]} & \mathcal{A}_{R2}^{[3]} & 0 & 0 & \cdots & 0 \\ \mathcal{A}_{L1}^{[4]} & \mathcal{A}_M^{[4]} & \mathcal{A}_{R1}^{[4]} & \mathcal{A}_{R2}^{[4]} & 0 & \cdots & 0 \\ \mathcal{A}_{L2}^{[5]} & \mathcal{A}_{L1}^{[5]} & \mathcal{A}_M^{[5]} & \mathcal{A}_{R1}^{[5]} & \mathcal{A}_{R2}^{[5]} & \cdots & 0 \\ 0 & \mathcal{A}_{L2}^{[6]} & \mathcal{A}_{L1}^{[6]} & \mathcal{A}_M^{[6]} & \mathcal{A}_{R1}^{[6]} & \mathcal{A}_{R2}^{[6]} & \cdots \\ 0 & 0 & \mathcal{A}_{L2}^{[7]} & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & \mathcal{A}_{L2}^{[n-2]} & \cdots & \mathcal{A}_M^{[n-2]} \end{bmatrix}, \tag{17}$$

with x_k given by (14). As in [1], the parametrization $A(x)x$ of $f(x)$ is not unique.

5 Simulation Results for 1D Advection

Figure 1 uses $n=44$ cells with $x_k \in \mathbb{R}^{400}$. The width Δr_j of cell j and time step $t_s(k)$ in (12) are varied to ensure stability. The initial states are determined by an empirical model relating the neutral densities and temperature to the integrated solar flux approximation ($F_{10.7}$) and activity level (Ap) [5]. Hence, initial conditions such as initial neutral and ion densities are varied through $F_{10.7}$.

The states of the truth model x_0 are initialized with $F_{10.7}=210$, which is used throughout the simulation. We assume that noise-free measurements of the number density of electron and the ion velocity in 40th cell are available so that

$$y_k = C(x_k) x_k, \tag{18}$$

where $C(x_k) \in \mathbb{R}^{2 \times 400}$ has entries 0's, 1's and $\frac{\exp(U_{l_0+l}(k))}{U_{l_0+l}(k)}$ in which $l_0 = 390, l = 1, \dots, 9$. Next, we assume the initial state x_0 is unknown and estimate x_k

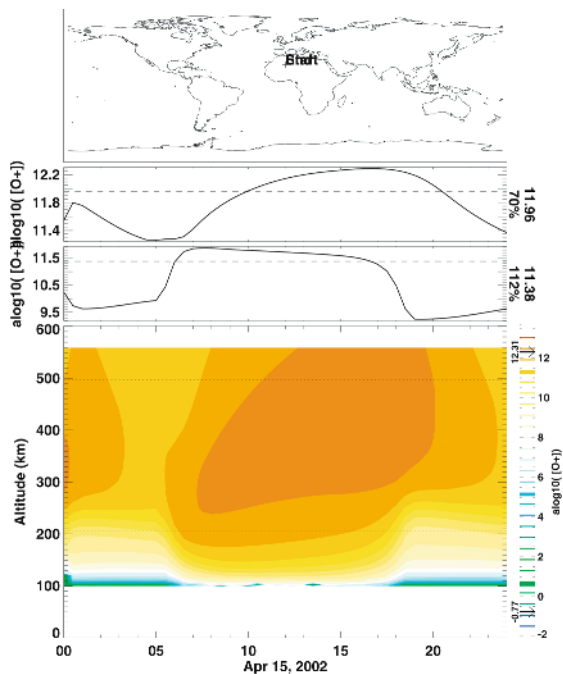


Fig. 2. Distribution of O^+ ions at altitudes above the location marked by '+' on the map (top plot). The second plot shows the log of the number density of O^+ along the upper dotted line in the bottom plot (around 500 km altitude), while the third plot shows the log of the number density of O^+ along the lower dotted line in the bottom plot (around 200 km altitude). The bottom plot shows the log of the number density of O^+ as a function of time and altitude. All logs are base 10.

using y_k . To do this, the estimator state is initialized with $F_{10.7} = 200$ so that $x_0^f \neq x_0$. The estimates \hat{x}_k are obtained by using the SDRE filter, where P_k is propagated using $P_0 = I_{400}$, $Q_k = I_{400}$, $R_k = I_2$, $C_k = C(x_k^f)$ and $A_k = A(x_k^f)$; $A(x_k^f)$ is given by (17) with x_k replaced by x_k^f . Although we set $F_{10.7} = 200$ at $k = 0$ to initialize the estimator, we set $F_{10.7} = 210$ for all $k > 0$.

Figure 2 shows the distribution of O^+ at altitudes above the location marked by '+', obtained by using the truth model initialized with $F_{10.7}=210$. The number density and ion velocity of O^+ at cells 30 and 40 of the truth model are shown in Figure 3. The estimates of ion density and ion velocity obtained from the SDRE filter by using measurements from cell 40 are also shown in the same figures. Finally, estimates of the ion density and ion velocity of O^+ obtained when no data assimilation is performed are also shown. Since the initial condition of the estimator x_0^f differs from the initial condition of the truth model x_0 , the number density and the ion velocity when no data assimilation is performed differs from the number density and the ion velocity of the truth model. However, when measurements are used for data assimilation, the SDRE-based filter provides good estimates of the states of the truth model.

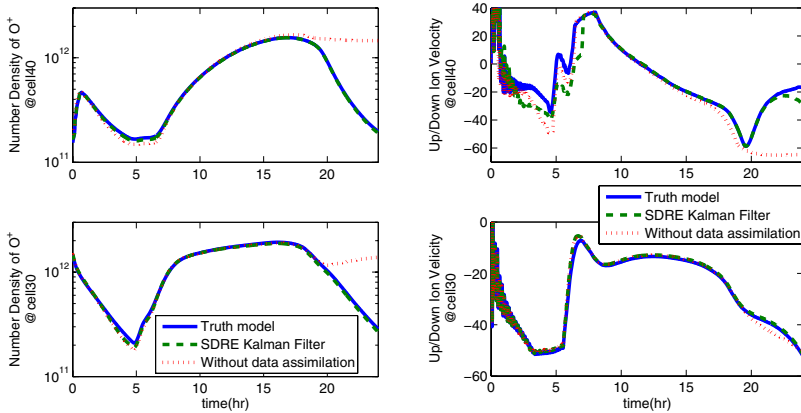


Fig. 3. Number density and ion velocity of O^+ at cells 30 and 40 are indicated by the solid lines. The estimates obtained when no data assimilation is performed are indicated by the dotted lines. These estimates differ from the state of the truth model because of the different initial conditions. The dashed lines indicated the SDRE filter estimates obtained when measurements from cell 40 are used for data assimilation. Note that after transients due to the erroneous initial conditions, the SDRE filter asymptotically achieves correct state estimates.

References

1. J. Chandrasekar, D. S. Bernstein, and A. Ridley. SDRE and EKF-based State Estimation for Two-Dimensional MHD Flow. 2006. preprint.
2. J. Chandrasekar, A. J. Ridley, and D. S. Bernstein. An SDRE-Based Asymptotic Observer for Nonlinear Discrete-Time Systems. In *Proc. Amer. Contr. Conf.*, pages 3630–3635, Portland, OR, June 2005.
3. G. Evensen. Advanced Data Assimilation for Strongly Nonlinear Dynamics. *Monthly Weather Rev.*, 125:1342–1354, 1997.
4. S. Gillijns, O. Barrero Mendoza, J. Chandrasekar, B. L. R. De Moor, D. S. Bernstein, and A. Ridley. What Is the Ensemble Kalman Filter and How Well Does it Work? In *Proc. Amer. Contr. Conf.*, Minneapolis, MN, June 2006.
5. A. J. Ridley, Y. Deng, and G. Toth. The Global Ionosphere-Thermosphere Model (GITM). *J. Atmos. Solar-Terrest. Phys.*, 2006. to appear.
6. R.G. Roble, E.C. Ridley, and R.E. Dickinson. On the Global Mean Structure of the Thermosphere. *J. Geophys. Res.*, 92:8745, 1987.
7. R.W. Schunk and A.F. Nagy. *Ionospheres*. Cambridge Press, Cambridge University, 2000.

Amplitude-Position Formulation of Data Assimilation*

Sai Ravela

Earth, Atmospheric and Planetary Sciences &
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
ravela@mit.edu

Abstract. Classical formulations of data-assimilation perform poorly when forecast locations of weather systems are displaced from their observations. They compensate position errors by adjusting amplitudes, which can produce unacceptably “distorted” states, adversely affecting analysis, verification and subsequent forecasts. It is non-trivial to identify sources of position error, but correcting misplaced forecasts is essential for operationally predicting strong, localized weather events such as tropical cyclones. In this paper, we propose a method that accounts for both position and amplitude errors. The proposed method assimilates observations in two steps. The first step is *field alignment*, where the current model state is aligned with observations by adjusting a continuous field of local displacements, subject to certain constraints. The second step is amplitude adjustment, where contemporary assimilation approaches are used. Our method shows improvements in analyses, with sparse and uncertain observations.

1 Introduction

Environmental data assimilation is the methodology for combining imperfect model predictions with uncertain data in a way that acknowledges their respective uncertainties. Data assimilation can be used to continually correct imperfect forecasts as well as guide a dynamic data acquisition procedure by indicating where and when new observations will be most helpful. However, data assimilation can only work when the estimation process properly represents all sources of error.

The difficulties created by improperly represented error are particularly apparent in mesoscale meteorological phenomena such as thunderstorms, squall-lines, hurricanes, precipitation, and fronts. Errors in mesoscale models can arise in many ways but they often manifest themselves as errors in the position of these features. Such a problem can be seen in Figure 1, which shows predicted and observed tracks for hurricane KATE. The various operational forecasts shown of the left side of this figure all contain significant position errors, giving trajectories that move in completely different directions. We would like to use data

* This material is supported in part by NSF ITR 0121182 and DDDAS 0540259.

assimilation to dynamically correct inaccurate predictions such as the hurricane KATE trajectories. But we typically don't know where these errors arise or exactly how they affect position. If we could attribute position error to a single (or even a small number) of incorrectly specified model inputs we could adjust these inputs using a classical parameter estimation procedure. It is, however, more likely that the position errors we observe are the aggregate result of errors in parameter values, initial conditions, and boundary conditions.

Since it is generally not possible to isolate the sources of position error in this way operational forecasters resort to ad hoc position adjustment procedures. One such procedure is commonly called *bogussing* [4], which removes the incorrectly positioned cyclone forecast and inserts a new "bogus" cyclone at the correct location, estimated from satellite imagery. Bogussing seems simple but it creates a host of problems. The bogus cyclone can introduce significant shocks in the dynamical system and if the original incorrectly positioned cyclone is not properly removed the resulting "ghost" vortex can severely degrade the track forecast.

A more sophisticated alternative to bogussing is to use data assimilation methods. Unfortunately, sequential [6], ensemble-based [5] and variational [9, 2] state estimation methods used in data assimilation applications adjust amplitudes to deal with position error. Adjusting amplitudes doesn't really fix position error, and instead, can produce unacceptably distorted estimates. These can, in turn, adversely affect the quality of subsequent forecasts [8, 10].

To see why this is, consider a Bayesian formulation of the data assimilation problem [7]. The state vector X_n at a discrete time t_n can be estimated using all measurements $Y_{0:n}$ from an associated conditional probability density $P(X_n|Y_{0:n})$. If we suppose that the dynamics is Markov in time, that is $P(X_n|X_{0:n-1}) = P(X_n|X_{n-1})$ and the observation errors are uncorrelated in time, and therefore $P(Y_{0:n}) = \prod_{i=0}^n P(Y_i)$, then $P(X_n|Y_{0:n})$ can be evaluated as:

$$P(X_n|Y_{0:n}) \propto P(Y_n|X_n)P(X_n|Y_{0:n-1}).$$

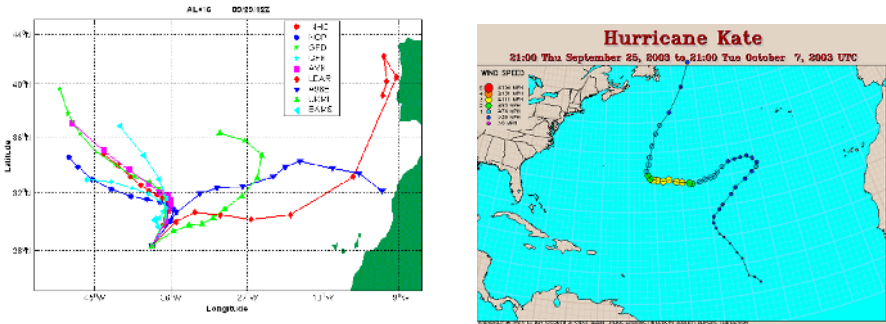


Fig. 1. The left picture shows forecasts of hurricane KATE launched on 09/29/2003 at 12Z using several operational models. The right picture shows its true path. The errors in positions are significant.

To simplify the ensuing discussion, we adopt a linear observation model, $Y_n = HX_n + \eta$, where η is additive measurement noise and uncorrelated in space and time, and H is a linear observation operator. Further, we drop the explicit dependence on time by writing $X = X_n$ and observation $Y = Y_n$, and the forecast $X^f = X_n|Y_{0:n-1}$. Then, if the distributions in question are assumed Gaussian, with $P(X^f) \sim N(X, B)$ and $\eta \sim N(0, R)$, the mean of $P(X_n|Y_{0:n})$ is equal to its mode, which is the value of X obtained from the solution: $X = X^f + BH^T(HBH^T + R)^{-1}(Y - HX^f)$.

This equation can be interpreted deterministically, where Y and X^f are fixed vectors called the observation and first guess, and X is the estimated state. The matrices B and R are the respective uncertainties in state and observations. As shown, these equations are a simplified version (due to the linear observation operator) of what is commonly known as "3DVAR" in the meteorological community [2, 9].

It can also be interpreted probabilistically, which forms the basis for the Ensemble Kalman Filter [5]. In this case, an ensemble of estimates at time t_{n-1} are forecast to time t_n using the model. Let us call the forecast ensemble $A^f = [X_1^f \dots X_S^f]$, where the columns of A^f are the S replicates of the ensemble. To implement the Ensemble Kalman Filter, we will assume that observation equation is linear (as before) with spatially and temporally uncorrelated additive noise. We let Z represent a matrix of perturbed observations, \tilde{A}^f be the deviation from mean \bar{A}^f of A^f , and write: $A = A^f + \tilde{A}^f(H\tilde{A}^f)^T \left[(H\tilde{A}^f)(H\tilde{A}^f)^T + R \right]^{-1} (Z - HA^f)$.

Both these methods perform equally poorly in the presence of position errors, because position errors introduce bias and/or degrade the forecast amplitude error-covariance. To see this, suppose that we have forecasts X_i , and $i = 1 \dots S$, which we deem a perfect Gaussian ensemble. Without loss of generality, we will represent this ensemble by a random variable X . We think of X as a spatially one-dimensional vector quantity, also referenced by a position variable p as $X(p)$. Let us suppose that $X \sim N(\bar{X}, B)$. That is, the covariance $B = E_X[\tilde{X}\tilde{X}^T]$, where E_X is the expectation under the distribution of X and $\tilde{X} = X - \bar{X}$. Let us consider a scalar position perturbation λ to this perfect forecast distribution, that is $X(p + \lambda)$. We now expand this via Taylor series (assuming the expansion is possible) as $X(p + \lambda) = X(p) + \lambda \frac{\partial X}{\partial p}$. The amplitude mean under the position perturbation can be written as $\bar{X}_\lambda = E_X[X(p + \lambda)] = \bar{X} + \lambda E_X[\frac{\partial X}{\partial p}]$. Further, we define the deviation of the gradients as $\Delta = \frac{\partial X}{\partial p} - E[\frac{\partial X}{\partial p}]$. Then the covariance as a result of the position perturbation is $C = B + \lambda E_X[\tilde{X}\Delta^T] + \lambda E_X[\Delta\tilde{X}^T] + \lambda^2 E_X[\Delta\Delta^T]$. If λ is assumed to be a Gaussian random variable with mean 0 and standard deviation σ_λ , then we can say that the expected amplitude covariance under the position perturbation is $E_\lambda[C] = B + \sigma_\lambda^2 E_X[\Delta\Delta^T]$. We define $C_\lambda = E_\lambda[C]$ and define $C_{\Delta\Delta} = E_X[\Delta\Delta^T]$ and therefore, we have $C_\lambda = B + \sigma_\lambda^2 C_{\Delta\Delta}$.

This equation says that the expected amplitude-covariance as a result of random position perturbations is less certain or "inflated" than without. Thus it is immaterial whether position errors show up in ensemble forecasts as a result of

model error, or they are introduced to account for model error in the forecast uncertainty. In either case, it weakens the forecast uncertainty, causing poor estimates. Therefore, rather than try to deal with position error indirectly, only through adjustments of amplitudes, we must also adjust positions! In particular, we can determine how the values predicted at model grid points need to be adjusted in amplitude and moved in space in order to achieve a better fit to observations. This perspective is the key our approach. The solution is general and applies equally well to meteorological, oceanographic, and hydrological applications.

2 Data Assimilation by Field Alignment

We reformulate the classical assimilation objective to allow position and amplitude adjustments. To make this framework more explicit it is useful to introduce some notation. Let $X = X(\mathbf{r}) = \{X[r_1^T] \dots X[r_m^T]\}$ be the model-state vector defined over a spatially discretized computational grid Ω , and $\mathbf{r}^T = \{r_i = (x_i, y_i)^T, i \in \Omega\}$ be the position indices. Similarly, let \mathbf{q} be a *vector* of displacements. That is, $\mathbf{q}^T = \{q_i = (\Delta x_i, \Delta y_i)^T, i \in \Omega\}$. Then the notation $X(\mathbf{r} - \mathbf{q})$ represents *displacement* of X by \mathbf{q} . The displacement field \mathbf{q} is real-valued, so $X(\mathbf{r} - \mathbf{q})$ must be evaluated by interpolation if necessary.

In a probabilistic sense, we may suppose that finding (X, \mathbf{q}) that has the maximum a posteriori probability in the distribution $P(X, \mathbf{q}|Y)$ is appropriate. Using Bayes rule we obtain $P(X, \mathbf{q}|Y) \propto P(Y|X, \mathbf{q})P(X^f|\mathbf{q})P(\mathbf{q})$. As before, we assume a linear observation model with uncorrelated noise in space and time, and Markov dynamics. If we make a Gaussian assumption of the component densities, we can write:

$$P(Y|X, \mathbf{q}) = \frac{1}{(2\pi)^{\frac{n}{2}} |R|^{\frac{1}{2}}} e^{-\frac{1}{2}(Y - H X(\mathbf{r} - \mathbf{q}))^T R^{-1} (Y - H X(\mathbf{r} - \mathbf{q}))} \tag{1}$$

This equation is the data-likelihood term. It implies that the observations can be related using a Gaussian model to the displaced state $X(\mathbf{r} - \mathbf{q})$, where $X(\mathbf{r})$ is defined on the original grid, and \mathbf{q} is a displacement field. We use the linear observation model here, and therefore, $Y = H X(\mathbf{r} - \mathbf{q}) + \eta, \eta \sim N(0, R)$. We should emphasize here that the observation vector is fixed. It's elements are always defined from the original grid.

$$P(X^f|\mathbf{q}) = \frac{1}{(2\pi)^{\frac{n}{2}} |B(\mathbf{q})|^{\frac{1}{2}}} e^{-\frac{1}{2}(X(\mathbf{r} - \mathbf{q}) - X^f(\mathbf{r} - \mathbf{q}))^T B(\mathbf{q})^{-1} (X(\mathbf{r} - \mathbf{q}) - X^f(\mathbf{r} - \mathbf{q}))} \tag{2}$$

This equation defines the *amplitude prior*. Given a fixed displacement field \mathbf{q} and a forecast $X^f(\mathbf{r})$ defined on the original grid, it states that the forecast distribution is assumed to be Gaussian in the position-corrected space, even if it isn't in the uncorrected space. Once we assume Gaussian statistics in a position corrected space, it is immediately clear that the forecast statistics are conditioned

on the displacement field. In particular, its second moment, the covariance B , is dependent on \mathbf{q} . By simple analogy, if we associate an error covariance on the original forecast grid, this error covariance will have to be remapped when the forecast grid is deformed. Thus, we write the forecast covariance as $B(\mathbf{q})$.

$$P(\mathbf{q}) = \frac{1}{C} e^{-L(\mathbf{q})} \tag{3}$$

This equation specifies a *displacement prior*. This prior is constructed from an energy function $L(\mathbf{q})$ which expresses constraints on the displacement field. The proposed method for constructing L is drawn from the nature of the expected displacement field. Displacements can be represented as smooth flow fields in many fluid flows and often arise from systematic and large-scale *background flow* errors, fore example see [1]. Smoothness naturally leads to a Tikhonov type formulation [11] and, in particular, $L(\mathbf{q})$ is designed as a gradient and a divergence penalty term. These constraints, expressed in quadratic form are:

$$L(\mathbf{q}) = \frac{w_1}{2} \sum_{j \in \Omega} \mathbf{tr}\{[\nabla \underline{q}_j][\nabla \underline{q}_j]^T\} + \frac{w_2}{2} \sum_{j \in \Omega} [\nabla \cdot \underline{q}_j]^2 \tag{4}$$

In Equation 4, \mathbf{q}_j refers to the j^{th} grid index and \mathbf{tr} is the trace. Equation 4 is a *weak constraint*, weighted by the corresponding weights w_1 and w_2 . Note that the constant C can be defined to make Equation 3 a proper probability density. In particular, define $Z(\mathbf{q}) = e^{-L(\mathbf{q})}$ and define $C = \int Z(\mathbf{q}) d\mathbf{q}$. This integral exists and converges.

With these definitions of probabilities, we are in a position to construct an objective by evaluating the log probability. To develop a solution easily, we make use of a statistical representation of uncertainty, using an *ensemble of forecast states* (this assumption is relaxed later). So suppose that S samples $\mathbf{X} = X_1 \dots X_S$ are to be estimated along with associated displacements $Q = \mathbf{q}_1, \dots \mathbf{q}_S$, from S forecasts $X_s^f, s = 1 \dots S$. Let $\mathbf{p}_s = \mathbf{r} - \mathbf{q}_s$ and $\bar{X}^f = \frac{1}{S} \sum_{s=1}^S X_s^f(\mathbf{p}_s)$. The background error covariance is:

$$B_Q = B(\mathbf{X}^f; Q) = \frac{1}{S-1} \sum_{s=1}^S (X_s^f(\mathbf{p}_s) - \bar{X}^f)(X_s^f(\mathbf{p}_s) - \bar{X}^f)^T \tag{5}$$

The ensemble framework leads to a quadratic objective from the log-likelihood, which we can write as $J(\mathbf{X}, Q) = \frac{1}{S} \sum_{s=1}^S J_s(\mathbf{X}, Q)$, where J_s is defined as, for $s = 1 \dots S$

$$\begin{aligned} J_s(\mathbf{X}, Q) &= \frac{1}{2} (X_s(\mathbf{p}_s) - X_s^f(\mathbf{p}_s))^T B_Q^{-1} (X_s(\mathbf{p}_s) - X_s^f(\mathbf{p}_s)) \\ &\quad + \frac{1}{2} (Y_s - H X_s(\mathbf{p}_s))^T R^{-1} (Y_s - H X_s(\mathbf{p}_s)) \\ &\quad + L(\mathbf{q}_s) - \ln(|B_Q|) \end{aligned} \tag{6}$$

We propose a solution using Euler-Lagrange equations. These can be written as:

$$\frac{\partial J}{\partial \mathbf{q}_s} = (\nabla X_s|_{\mathbf{p}_s} - \nabla X_s^f|_{\mathbf{p}_s})^T B_{\hat{Q}}^{-1} (X_s(\mathbf{p}_s) - X_s^f(\mathbf{p}_s)) + \nabla X_s|_{\mathbf{p}_s} H^T R^{-1} (H X_s(\mathbf{p}_s) - Y_s) + \frac{\partial L}{\partial \mathbf{q}_s} = 0 \quad (7)$$

$$\frac{\partial J}{\partial X_s} = B_{\hat{Q}}^{-1} (X_s(\mathbf{p}_s) - X_s^f(\mathbf{p}_s)) + H^T R^{-1} (H X_s(\mathbf{p}_s) - Y_s) = 0 \quad (8)$$

These equations are highly non-linear functions in X_s and \mathbf{q}_s . We solve them sequentially, in exactly two steps. In the first step we fix X_s to X_s^f in Equation 7 and solve for \hat{Q} . We then use this solution to solve Equation 8. The sequential approach is a *two-step* approximation. The first-step is the the displacement or alignment equation, written as:

$$\frac{\partial L}{\partial \mathbf{q}_s} + \nabla X_s^{fT}|_{\mathbf{p}_s} H^T R^{-1} (H X_s^f(\mathbf{p}_s) - Y_s) = 0 \quad (9)$$

Using the regularization constraints the alignment equation at a node i now becomes:

$$w_1 \nabla^2 \underline{q}_{s,i} + w_2 \nabla(\nabla \cdot \underline{q}_{s,i}) + [\nabla X_s^{fT}|_{\mathbf{p}_s} H^T R^{-1} (H [X_s^f(\mathbf{p}_s)] - Y_s)]_i = 0 \quad (10)$$

Equation 10 is the field alignment formulation. It introduces a forcing based on the residual between the model- and observation-fields. The constraints on the displacement field allow the forcing to propagate to a consistent solution. Equation 10 is also non-linear, and is solved iteratively, as a Poisson equation. During each iteration \mathbf{q}_s is computed by holding the forcing term constant. The estimate of displacement at each iteration is then used to deform a copy of the original forecast model-field using bi-cubic interpolation for the next iteration. The process is repeated until a small displacement residual is obtained, the misfit with observations does not improve, or an iteration limit is reached. Upon convergence, we have an aligned forecast ensemble $X_s^f(\hat{\mathbf{p}}_s)$, $s = 1 \dots S$, from which $B_{\hat{Q}}$ is computed. With these quantities, the amplitude recovery is written as:

$$X_s(\hat{\mathbf{p}}_s) = X_s^f(\hat{\mathbf{p}}_s) + B_{\hat{Q}} H^T (H B_{\hat{Q}} H^T + R)^{-1} (Y_s - H X_s^f(\hat{\mathbf{p}}_s)) \quad (11)$$

Equation 11 can be implemented using several familiar schemes. Here we outline three:

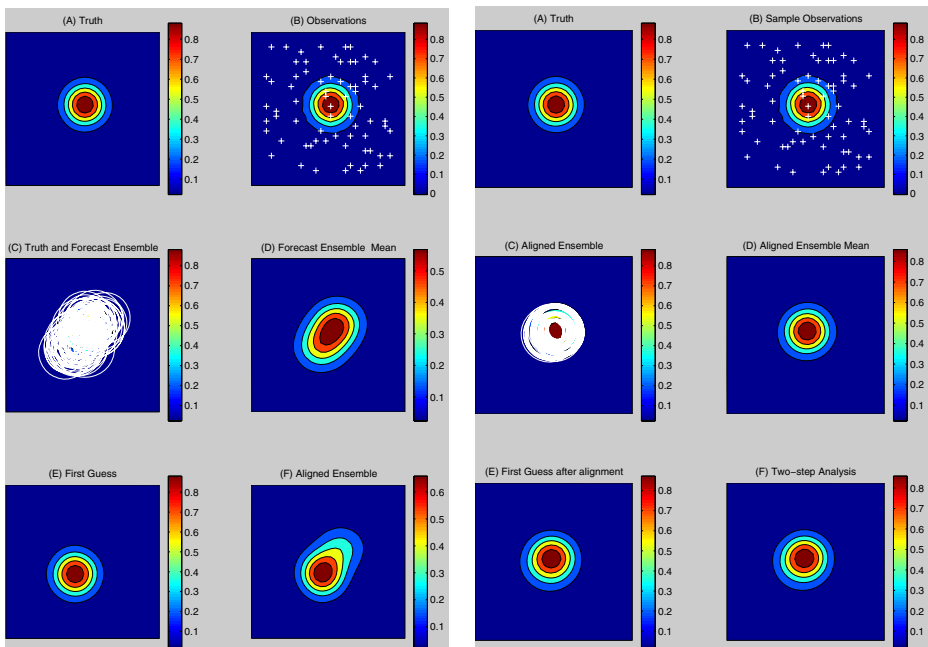
Ensemble Scheme: Equation 11 can be an ensemble Kalman filter, because \hat{Q} is fixed. Of course, several EnKF schemes can be implemented using Equation 11.

Deterministic Ensemble Scheme: The background can be computed from individual, aligned forecast replicates. Aligned replicates can be used to compute a statistical background error covariance. Once this is done, deterministic assimilation proceeds from the background.

Purely Deterministic Scheme: The proposed two-step method can also be used when a forecast ensemble is not available. Since the alignment does not depend on the background error covariance, one can align the first-guess field and use the aligned first-guess to craft a state-dependent background error covariance [3]. In essence, the same procedure that would be used with the unaligned first-guess can be also used with the aligned first-guess.

3 Example

Figure 3a(A) depicts the vorticity contours of a true vortex field of size 32×32 . Figure 3a(B) depicts observed locations (+), and the observation field is gen-



(a) 3DVAR: Panel (A) is a true vortex field, observed at sparse locations (+ sign) shown in panel (B). Panel (C) shows an overlay of the forecast ensemble (white rings). Panel (D) depicts the forecast ensemble mean. The first guess is selected to be a coherent vortex of similar structure as the observations from the ensemble, as shown in Panel (E). Panel (F) depicts the result of a deterministic assimilation scheme.

(b) Two-step method: Panel (A) and (B) are the same as Figure 3a. Panel (C) is the result of field alignment. This produces a better forecast ensemble mean shown in panel (D). The first guess is selected from the ensemble as shown in panel (E). Panel (F) depicts the result of a deterministic assimilation scheme after alignment.

Fig. 2. Comparison of 3DVAR and two-step approaches

erated by adding to truth a 1% (of peak amplitude) uncorrelated noise field in space. Only 6.8% of the state is observed (70 observations), at randomly selected points. In particular, about 12 observations lie on significant contour levels of the vortex.

Figure 3a(C) depicts the contours (white) of a forecast ensemble (25 ensemble members) overlaid on truth. They completely “cover” truth, suggesting that truth is well-captured by the ensemble members. However, all the ensemble members have position errors from truth as well as amplitude errors. Their peak amplitude can vary by as much as 10% and their position by 10 pixels. Figure 3a(D) depicts the forecast ensemble mean, which is broader than the true vortex. Figure 3a(E) depicts the first-guess used for a 3DVAR procedure. The background error covariance (B) is computed from the forecast ensemble, and the observation uncertainty is constructed from a 0.01 standard deviation, iid. Figure 3a(F) depicts the analysis produced by 3DVAR. The smearing of the vortex can be explained by the fact that position errors produce an artificially broad (or more uncertain) background error covariance.

Figure 3b depicts the same example using the proposed two-step approach. Figure 3b(A) and Figure 3b(B) are identical to corresponding plots in Figure 3a. Figure 3b(C) depicts the *aligned* ensemble, using the alignment formulation developed in the previous section. In particular, equation 10 is used to align each ensemble member with a perturbed observation (of which Figure 3b(B) is one sample). The ensemble mean after alignment is shown in Figure 3b(D). An ensemble member is selected as the first guess for 3DVAR, as shown in Figure 3b(E). The analysis is shown in Figure 3b(F). The reason why the two-step approach works is because the aligned ensemble may be considered to be a far better depiction of background uncertainty than the one containing position errors. Experiments using the EnKF to produce the analysis leaves us with the same conclusions, as the one-dimensional examples also show. These are not repeated here.

4 Conclusions

Our method is a Bayesian perspective of the assimilation problem that simultaneously considers position and amplitude errors. Our approach has several distinct advantages: (a) It is useful for a wide range of environmental assimilation problems because a more general description of error than current methods. (b) Our approach does not require features to be identified for correcting position errors. This is a significant advantage because features cannot always be clearly delineated from sparse observations and they are not explicitly defined in the Eulerian models most commonly used for environmental forecasting. (c) our approach can be integrated easily with current operational implementations, thereby making this effort more likely to have a real impact. (d) We have examined extensions of the two-step algorithm to multivariate fields and 3D fields. In the former case, we see that alignment does not perturb a preexisting dynamical balance. Finally, the regularization constraint in field alignment is a weak

constraint and the weights determine how strongly the constraints influence the flow field. The constraint in L is modeled as such because we expect the fluid flow to be smooth. From a regularization point of view, there can be other choices [12] as well.

References

1. G. D. Alexander, J. A. Weinman, and J. L. Schols. The use of digital warping of microwave integrated water vapor imagery to improve forecasts of marine extratropical cyclones. *Monthly Weather Review*, 126:1469–1495, June 1998.
2. P. Courtier. Variational methods. *J. Meteor. Soc. Japan*, 75, 1997.
3. R. Daley. *Atmospheric Data Analysis*. Cambridge University Press, 1994.
4. C. Davis and S. Low-Nam. The near-afwa tropical cyclone bogussing scheme. *Technical Memorandum, Air Force Weather Agency (AFWA), Omaha, NE*, [<http://www.mmm.ucar.edu/mm5/mm5v3/tc-report.pdf>], 2001.
5. G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53:342–367, 2003.
6. A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
7. A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
8. C. D. Jones and B. MacPherson. A latent heat nudging scheme for assimilation of precipitation data into an operation mesoscale model. *Meteorol. Appl.*, pages 269–277, 1997.
9. A. C. Lorenc. Analysis method for numerical weather predictin. *Q. J. R. Meteorol. Soc.*, 112:1177–1194, 1986.
10. H. J. Thiebaut, P. R. Julian, and G. J. DiMego. Areal versus collocation data quality control. In *Intl. Symp. on Assimilation of Observations in Meteorolgy and Oceanography*, pages 255–260, Clermont-Ferrand, France, WMO, 1990.
11. A.N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Wiley, New York, 1977.
12. G. Wabha and J. Wendelberger. Some new mathematical methods for variational objective analysis using splines and cross-validation. *Monthly Weather Review*, 108, 1980.

Detection of Tornadoes Using an Incremental Revised Support Vector Machine with Filters

Hyung-Jin Son and Theodore B. Trafalis

School of Industrial Engineering, The University of Oklahoma
202 W. Boyd, CEC 124, Norman, OK 73019, U.S.A.
{son, ttrafalas}@ou.edu

Abstract. Recently Support Vector Machines (SVMs) have played a leading role in pattern classification. SVMs are quite effective to classify static data in numerous applications. However, the use of SVMs in dynamically data driven application systems (DDDAS) is somewhat limited. This motivates the development of incremental approaches to handle DDDAS. In an incremental learning approach, it is critical to keep a certain number of support vectors (SVs) without seriously sacrificing the generalization performance of SVMs. In this paper a novel incremental SVM method, called an incremental revised support vector machine with filters (IRSVMF) is proposed to resolve the above limitations. Computational experiments with tornado data show that this approach is quite effective to reduce the number of SVs and computing time and to increase the detection rate of tornadoes.

1 Introduction

Support Vector Machines (SVMs) have played a leading role in pattern classification. Applying SVMs into the real world has motivated the development of incremental approaches to deal with huge data that are continuously coming to a learning system. Numerous publications point out that the standard SVMs cannot properly handle large-scale data sets and that the incremental approach is a remedy to overcome limitations of the standard SVMs [1, 2, 3].

In applying the SVMs approach in an incremental framework for classification problems, we will face several limitations as follows:

First, support vectors (SVs) are accumulated as the incremental learning process is repeated. Therefore, it is important to control the number of SVs in SVMs. Second, SVMs waste most computing time for computation of kernel function values using less important data. If SVMs are used for training data from a particular classification problem such as an unbalanced classification problem in which there are many data in one class (less important) and few data in the other class (important), then use of computing time for kernel function evaluation among data points that are less important should be avoided to reduce the training time.

The tornado detection problem is an application to be considered in this study. It can be characterized as follows: First, it is a two-class (tornado and non-tornado) classification problem. Second, it is a problem with unbalanced data. The tornado class

consists of few data in the entire tornadic and non-tornadic data set. Third, it is an asymmetric data importance problem. That is, the tornado class is relatively more important than the non-tornado class. Fourth, weather data related to tornado are periodically provided by weather radars. The standard SVM or other variants cannot properly train the weather data due to limited capacity of computation for training, and the size and periodic inflow of these data. Thus, the use of the standard SVM in dynamically data driven application systems (DDDAS) such as weather prediction is somewhat limited. This motivates the development of incremental approaches to handle DDDAS related to tornado detection.

Therefore the standard SVM should be revised to overcome these limitations. The objective of this study is (1) to develop a revised SVM to reduce the number of support vectors, (2) to construct an incremental learning procedure with the revised SVM, and (3) to make the incremental learning applicable to on-line settings by creating a filter to discard the most unimportant data.

This paper is organized as follows: Section 2 describes the standard SVM. In section 3, the incremental Revised SVM with filter (IRSVMF) is proposed. The tornado detection problem is described in section 4. Section 5 contains computational experiments and results. Conclusion follows in section 6.

2 Support Vector Machines

The basic idea of SVMs, introduced by Vapnik [4], is to construct a decision hyperplane to separate two-class samples maximizing the margin of separation. SVMs can be applied both in linearly separable and inseparable patterns cases. A brief mathematical explanation of SVMs is as follows:

Consider the training sample set $\{(x_i, y_i)\}_{i=1}^N$, where x_i is the input pattern for the i th sample, and y_i is the corresponding output (± 1). Assume that the training samples and corresponding outputs are provided (i.e., supervised learning). The aim of SVM is to obtain the optimal weight vector w_o and bias b_o for the decision hyperplane, $w_o^T x + b_o = 0$, by solving the following optimization problem.

$$\begin{aligned} & \text{Max } \frac{1}{2} \|w\|^2 & (1) \\ & \text{s.t. } y_i [w^T x_i + b] \geq 1, \quad i = 1, 2, \dots, N \end{aligned}$$

The data points (x_i, y_i) along the hyperplanes with equality shown in (1) are called support vectors (i.e., critical data points that are located in the closest positions to the decision hyperplane).

In the linearly inseparable patterns case where there are some infeasibilities for the constraints of (1), the objective function is defined as $\Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i$, where C is a user-defined parameter that controls the tradeoff between the number of inseparable data points and generalization of the SVM. Note that $\xi_i, i = 1, 2, \dots, N$ are slack variables that measure the deviation of a data point from the separating hyperplane.

Using duality theory and kernel method [5], we have the following constrained optimization problem:

$$\begin{aligned} \text{Max } F(\alpha) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{s.t. } \sum_{i=1}^N \alpha_i y_i &= 0 \\ 0 \leq \alpha_i &\leq C \text{ for } i = 1, 2, \dots, N \end{aligned} \quad (2)$$

Note that K is a kernel function satisfying Mercer's condition. Possible kernel functions are polynomial or radial basis functions [5]. The resulting decision function in a binary classification problem takes the form $f(x) = \text{sign}(\sum_{i=1}^N \alpha_i y_i K(x_i, x) + b)$, where α_i 's are the optimal solutions of (2), and b is computed using the Kuhn-Tucker conditions [4].

Bennett and Breadensteiner [6] developed the reduced convex hull concept giving a geometric explanation of the standard SVM. Crisp and Burges [7] also developed the same concept in a geometric interpretation of ν -SVM independently. The reduced SVM minimizes the distance of the reduced convex hulls of the positive and negative class respectively. This approach removes the effect of noisy data (e.g., outliers) and reduces the number of support vectors. In contrast, it increases the generalization error.

3 Incremental Revised Support Vector Machine with Filters

3.1 Revised Support Vector Machine

The standard SVM produces a huge amount of support vectors especially when the positive and negative training samples are highly overlapped with each other. It's obvious that the big size of support vectors requires more computing time and more storage space for training. Thus, it is natural to say that reducing the computational time and cost of the SVM is equivalent to decreasing the number of support vectors.

In addition, noisy data (e.g., outliers) might significantly affect the standard SVM producing an incorrect decision function. As a result, the incorrect decision function generates unexpected generalization errors and affects sequentially the next steps in the incremental learning process.

Therefore, the standard SVM must be modified to resolve the above problems. The geometric interpretation of the reduced SVM is quite useful and provides an alternative for modifying the standard SVM. However, it might be inefficient for particular problems such as unbalanced problems in which there are huge differences of data sizes in two classes. A similar situation happens in asymmetric importance problems where there are few data in one class with important meaning and a lot of unimportant data in the other class. For example, tornadic data are very few (as little as 2%) relatively to nontornadic data. If the reduced SVM is applied to this tornado detection problem, some valuable tornadic data will be lost. Thus, in order to properly solve this problem, the following geometric concept is proposed.

Consider a two-class classification problem. One class has few but important data. The other class has a lot of unimportant data. The data in the important class is preserved, and the data in the unimportant class will be reduced.

Geometrically, the revised SVM optimization problem in the linearly inseparable case takes the following form:

$$\begin{aligned}
 \text{Min} \quad & \frac{1}{2} \left\| \sum_{i=1}^{N_P} \alpha_i x_i - \sum_{j=1}^{N_N} \beta_j x_j \right\|^2 & (3) \\
 \text{s.t.} \quad & \sum_{i=1}^{N_P} \alpha_i = 1 ; \quad \sum_{j=1}^{N_N} \beta_j = 1 \\
 & 0 \leq \alpha_i \leq 1 \quad \text{for } i = 1, 2, \dots, N_P \text{ (important class)} \\
 & 0 \leq \beta_j \leq \mu \quad \text{for } j = 1, 2, \dots, N_N \text{ (unimportant class)} \\
 & \text{where } 0 < \mu < 1
 \end{aligned}$$

By imposing an upper bound on each multiplier, β_j , the convex hull that consists of data points in the unimportant class is shrunk. In contrast, the convex hull in the important class is preserved. The resulting revised SVM optimization problem for the linearly inseparable case is to maximize (4) subject to the constraints of (3).

$$Q(\alpha, \beta) = \sum_{i=1}^{N_P} \sum_{j=1}^{N_N} \alpha_i \beta_j K(x_i, x_j) - \frac{1}{2} \left[\sum_{i=1}^{N_P} \sum_{j=1}^{N_P} \alpha_i \alpha_j K(x_i, x_j) + \sum_{i=1}^{N_N} \sum_{j=1}^{N_N} \beta_i \beta_j K(x_i, x_j) \right] \quad (4)$$

3.2 Incremental Revised Support Vector Machine with Filters

In this study, we utilize the revised SVM instead of the standard SVM for training data in the incremental learning process. A filter is created to remove the most likely unimportant data prior to training.

If the whole data set is available, it is divided into several batches. In an on-line setting, only the first batch is created. If historic data are available, the first batch can be replaced by these data. In literature (e.g., [1]), the size of the batch is determined arbitrarily. The appropriate batch size can be obtained considering a trade-off between generalization error rate and computing time. For details, refer to [8]. Data in the first batch is trained by the revised SVM identifying the support vectors. These support vectors are included in the second batch. The iterative procedure is repeated until all batches are trained. After all data are trained, the final decision function is made for classification. The supporting hyperplane defined through the support vectors of the unimportant class plays a role as a filter. Because most of the unimportant data are located on the side of the supporting hyperplane referring to the unimportant class, this supporting hyperplane is a good yardstick for removing the possible unimportant data before training.

Data points passing this filter are put in the next batch until the size of the batch is filled up to the predetermined batch size. This filter is updated for every batch. Thus, this approach requires fewer batches than the traditional batch method in the incremental learning procedure. Hence, a new algorithm for incremental learning with revised SVM and a filter (IRSVMF) is proposed as follows:

- Step 1. Determine the optimal batch size based on generalization error rate and computing time.
- Step 2. Train the data in the first batch by the revised SVM.
- Step 3. Add data points representing support vectors obtained in step 2, in the next batch.
- Step 4. Inject a new point to the filter.
- Step 5. If a new data point is located on the negative side of the supporting hyperplane of the reduced convex hull of the unimportant class, then discard the data point. Otherwise, add it in the next batch.
- Step 6. If data size in a batch is equal to the predetermined batch size, go to the next step. Otherwise, go to step 4.
- Step 7. Train the data in the batch and obtain support vectors, and corresponding decision function.
- Step 8. Update the filter. Go to step 3.

Advantages of the incremental learning with the revised SVM and filter are as follows: First, it makes a selected support vector set to be as small as possible. Second, it can keep the memory and time complexity of the learning algorithm at a manageable level. Third, it can predict at a time when the whole data set is not yet available (on-line setting). When the data set is only periodically available (e.g., weather data or financial data), the traditional learning approach should wait until all data are available for training. In contrast, the incremental approach can be applied to train a small portion of the whole data set and has a capability to predict the class of an incoming data point using the constructed classifier before the next data point has arrived. Fourth, it can remove the serious effect of noisy data (e.g., outliers).

4 Tornado Detection

In real world, the SVM concept has been applied to many areas such as image classification, bioinformatics, text-categorization, data mining, and meteorology (e.g., [9]). It's hard to use the standard SVM due to many limitations (e.g., memory requirement, timely manner, availability of input data). Thus, the incremental learning approach developed in this study is quite suitable to the following situations: First, when all information data in a system cannot be obtained at once where periodically information data are injected to a system. Second, when one class has relatively more important points than the other one in a two-class problem.

One interesting application of the above incremental learning approach with good performance is the tornado detection problem. Tornado is a rare but significantly critical event in the real world as well as in the meteorology community. Based on the weather data produced from the Weather Surveillance Radar 1988 Doppler (WSR-88D), the Mesocyclone Detection Algorithm (MDA) is currently used to detect tornados [10]. Typically, a tornado confusion matrix is utilized in order to measure the performance for tornado detection as shown in Table 1.

Table 1. Tornado Confusion Matrix

Forecast Tornado	Tornado Observed			
		Yes	No	
	Yes	Hit (a)	False alarm (b)	
No	Miss (c)	Correct (d)	“No” Forecasts	
	“Yes” Observation	“No” Observation	Total number of observations	

When the revised SVM is used, the “false alarm” rate might be increased because the size of the convex hull containing non-tornado data is reduced. However, the number of “miss” events will be substantially reduced because the convex hull containing tornado data is not shrunk. It is critical to decrease the “miss” rate in the tornado detection problem because a high “miss” rate brings unexpected and serious disasters frequently. From the confusion matrix, the probability of detection (POD) is computed in (5).

$$\text{Probability of detection (POD)} = \frac{a}{a+c}. \quad (5)$$

5 Experiments and Computational Results

5.1 Experiments

In this study, we use the MDA data provided by the National Severe Storms Laboratory (NSSL). These tornadic and nontornadic data generated from 1994 to 1999 are randomly selected to produce ten training sets and ten testing ones where each set has 1500 data. Each datum has 23 attributes that are related to information such as velocity and shear. These attributes have been successfully used for tornado detection in the literature [10]. To reflect the real situation, tornadic data form 10% in a training set as well as in a testing one.

Three approaches are performed and compared: the incremental approach with standard SVM (ISSVM), the incremental approach with revised SVM (IRSVM), and the incremental approach with revised SVM and filter (IRSVMF). These approaches are compared in terms of POD, total CPU time, number of batches, and “miss” rate. For each approach, training and testing are performed ten times respectively, and their average values are computed in terms of the above criteria.

In the incremental approaches with standard SVM and with revised SVM, each batch size is set to 300 (for details, refer to [8]). The incremental approach with filters uses a batch with size of 300 data, which consists of data passing the filter. Note that support vectors obtained from the previous batch are added to the training batch. MATLAB codes provided by [11] are entirely revised to run the incremental step and used in a Pentium IV 2.8GHz with 1 GB RAM to perform all experiments.

5.2 Computational Results

After performing each approach with ten training and testing data sets, the averaged results are shown in Table 2.

Table 2. Comparisons of methods

Methods	POD	Total CPU time (Sec)	Number of SVs	“Miss” rate (%)
ISSVM	0.62	754.62	57	3.83
IRSVM	0.69	406.26	11	3.14
IRSVMF	0.60	314.46	11	3.97

Even if the total CPU time and the number of support vectors are reduced, IRSVM outperforms ISSVM in terms of POD and “miss” rate. The CPU time and the number of support vectors for each batch are shown in Figure 1.

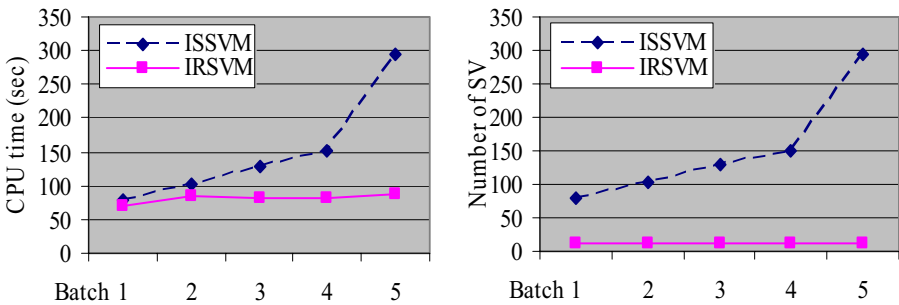


Fig. 1. Comparison of ISSVM and IRSVM in terms of computing time and number of SVs

Since the number of support vectors is increased as each batch is sequentially trained, the computing time is also dramatically increased, whereas IRSVM keeps the same number of support vectors and requires smaller computing time. When IRSVMF is applied, computing time is also significantly reduced although POD is slightly dropped.

6 Conclusions

Tornados are rare but very critical events in real world as well as in the meteorological community. It is quite important that large-scale data such as weather radar data should be properly handled such that the accuracy of tornado prediction should be improved. To accomplish those goals, the incremental revised SVM with a filter concept is presented. Our results show that the revised SVM outperforms the standard SVM in terms of computing time and number of batches and can be used in other DDDAS. The revised SVM concept improves the accuracy of tornado prediction, and

the use of filter in the revised SVM also reduces the computing time. In the future, those algorithms will be tested in a real operational setting.

Acknowledgement

This material is based on research funded by National Science Foundation Grant EIA-0205628.

References

1. Demeniconi, C. and D. Gunopulos, 2001. "Incremental support vector machine construction," *Proceedings of the IEEE International Conference on Data Mining*, San Jose, CA, pp. 589-592.
2. Syed, N. A., H. Liu, and K. K. Sung, 1999. "Incremental Learning with Support Vector Machines," *Workshop on Support Vector Machines, International Joint Conference on Artificial Intelligence*, Stockholm, Sweden.
3. Shilton, A., M. Palaniswami, D. Ralph, and A. C. Tsoi, 2005. "Incremental training of support vector machines," *IEEE Transactions on neural networks*, vol. 16, no. 1, pp 114-131.
4. Vapnik, V. N., 1995. *The Nature of Statistical Learning Theory*. New York, NY: Springer Verlag.
5. Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*, 2nd edit., Upper Saddle River, NJ: Prentice-Hall.
6. Bennet, K. P, and E. J. Bredensteiner, 2000. "Duality and Geometry in SVM Classifiers," *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 57-64.
7. Crisp, David J. and C. J. C. Burges, 1999. *A Geometric Interpretation of ν -SVM Classifiers*. In *Advances in Neural Information Processing Systems (NIPS)* vol. 12. Cambridge, MA: MIT Press.
8. Son, H, T. B. Trafalis, and M. Richman, 2005. "Determination of the Optimal Batch Size in Incremental Approaches: An Application to Tornado Detection," *Proceedings of the International Joint Conference of Neural Network*, Montreal, Canada, pp. 2706-2710.
9. Cristianini, N. and J. Shawe-Taylor, 2000. *An Introduction to support vector machines*. Cambridge University Press.
10. Marzban, C. and G. J. Stumpf, "A Neural Network for Tornado Prediction Based on Doppler Radar Derived Attributes," *Journal of Applied Meteorology*, 1996, vol. 35, pp. 617-626.
11. Gunn, S. R., 1997. *Support Vector Machines for Classification and Regression*. Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton.

A Generic Multi-scale Modeling Framework for Reactive Observing Systems: An Overview*

Leana Golubchik, David Caron, Abhimanyu Das, Amit Dhariwal,
Ramesh Govindan, David Kempe, Carl Oberg, Abhishek Sharma,
Beth Stauffer, Gaurav Sukhatme, and Bin Zhang

University of Southern California, Los Angeles, CA 90089
`leana@cs.usc.edu`

Abstract. Observing systems facilitate scientific studies by instrumenting the real world and collecting corresponding measurements, with the aim of detecting and tracking phenomena of interest. A wide range of critical environmental monitoring objectives in resource management, environmental protection, and public health all require distributed observing systems. The goal of such systems is to help scientists verify or falsify hypotheses with useful samples taken by the stationary and mobile units, as well as to analyze data autonomously to discover interesting trends or alarming conditions. In our project, we focus on a class of observing systems which are *embedded* into the environment, consist of *stationary and mobile* sensors, and *react* to collected observations by reconfiguring the system and adapting which observations are collected next. In this paper, we give an overview of our project in the context of a marine biology application.

1 Introduction

Observing systems facilitate scientific studies by instrumenting the real world and collecting corresponding measurements, with the aim of detecting and tracking phenomena of interest. In our project, we focus on a class of observing systems which are (1) *embedded* into the environment, (2) consist of *stationary and mobile* sensors, and (3) *react* to collected observations by reconfiguring the system and adapting which observations are collected next. We refer to these as Reactive Observing Systems (ROS). The goal of ROS is to help scientists verify or falsify hypotheses with useful samples taken by the stationary and mobile units, as well as to analyze data autonomously to discover interesting trends or alarming conditions.

We explore ROS in the context of a marine biology application, where the system monitors, e.g., water temperature and light as well as concentrations of micro-organisms and algae in a body of water. Using a hybrid network of

* This research has been funded by the NSF DDDAS 0540420 grant. It has also been funded in part by the NSF Center for Embedded Networked Sensing Cooperative Agreement CCR-0120778, the National Oceanic and Atmospheric Administration Grant NA05NOS47812228, and the NSF EIA-0121141 grant.

stationary and mobile sensors, communicating both via wired and wireless links, the system collects fine-grained measurements of interesting information in near real-time. An example use of such a system is the rapid identification of microorganisms to predict the onset of algal blooms. Such blooms can have devastating economic consequences, as recently seen in [4].

However, current technology (and any realistic prediction of technologies in the near future) precludes sampling all possibly relevant data. For instance, bandwidth limitations between the stationary sensors make it impossible to collect all of the sensed data. Similarly, time and storage capacity constraints for the mobile entities severely curtail the number and locations of samples they can take.

To make good use of the limited resources, we need to develop a framework for ROS capable of optimizing and controlling the set of samples to be taken at any given time, taking into consideration the application's objectives and system resource constraints. To support such an optimization and control process, a significant part of the framework must be dedicated to the development of models of data, and their automatic validation¹ or adaptation. As part of the validation and adaptation process, the framework must also include a distributed support mechanism for locating data of interest. We refer to this framework as AMBROSia (Autonomous Model-Based Reactive Observing System).

We seek to develop AMBROSia as a multi-scale modeling framework for ROS. AMBROSia allows applications to construct inter-related models of varying spatio-temporal scope based on collected data. Guided by the models, the reactive elements of the system predict where interesting data and phenomena are likely to be found. In the process of constructing models, the system actively seeks most useful data to improve both, the models and phenomenon detection and tracking. In a feedback cycle, this data acquisition is guided by previous, perhaps less precise, models. Thus, AMBROSia enables optimal collection of measurements in a manner that respects system resource constraints, yet improves the overall fidelity of phenomenon detection and tracking.

The system we propose to develop is targeted at the marine application outlined above, and described in more detail below. However, we believe that many of the components we develop, as well as the general AMBROSia framework, may be quite useful in other settings.

2 Monitoring Marine Ecosystems

A wide range of critical environmental monitoring objectives in resource management, environmental protection, and public health all require distributed observing systems. Here we focus primarily on a marine biology application. Our application's primary long term scientific goal is to understand, and ultimately predict, the conditions under which specific populations of marine microorganisms develop in nature. A fundamental requirement for attaining this objective is the correlation of environmental conditions with microorganismal abundances at

¹ By validation we mean a process of verifying the accuracy of models, based on collected data, and subsequent discarding or updating/adaptation of those models.

the small spatial and temporal scales that are relevant to the organisms. This is not possible with current technology and methodological approaches. Sampling the environment with high resolution and identifying microorganisms in situ in near-real time will constitute a revolutionary advance in the study of the ecology of marine microbial species. In addition, the rapid identification of aquatic microorganisms will be extremely valuable for the early detection of harmful organisms and the mitigation of their effects on the environment and the human population.

Marine microorganisms such as viruses, bacteria, microalgae, and protozoa have a major impact on the ecology of the coastal ocean. For example, blooms of harmful and/or toxic algae (e.g., red, brown and green tides) in aquatic ecosystems have increased dramatically on a global scale in recent years [1, 2]. These events result in the loss of human life each year, and economic losses in the billions of dollars due to effects on fisheries and tourism. Likewise, the increasing encroachment of humans along coasts has resulted in the recognition of potential public health issues as a consequence of the introduction of pathogenic microorganisms into these waters from land runoff, storm drains and sewage outflow. Similar concerns exist regarding the potential for contamination of drinking water supplies with harmful, pathogenic or nuisance microbial species. Today, the environmental factors that stimulate the growth of such microorganisms are still poorly understood, and tests for their abundances are not sufficiently rapid to detect the onset of major outbreaks.

We now give a brief illustration of the types of hypotheses which are of interest, i.e., this serves as an example of a potential AMBROSia application. Of course, our goal is not to evaluate these specific hypotheses, but rather to design and build a system capable of aiding scientists in the evaluation of hypotheses.

A popular hypothesis to explain accumulations of harmful microalgae near the shore (resulting in ‘red’ tides) includes the release of cysts from the sediments, growth in the water column, and then accumulation near shore as a result of favorable weather conditions. Thus, [6] suggests that a combination of winds — specifically wind speed, direction and duration that result in transport of the population towards the coast — may lead to red tides along the coast. Alternate theories posit the importance of upwelling events (the movement of deep water and nutrients contained in deep water into surface waters) or the breaking of internal waves that propagate along subsurface density discontinuities, as contributors to these coastal phenomena.

The Need for a Hybrid Sensing Approach. In tracking the phenomena described above, we encounter the challenge that the relevant locations can not be predicted at deployment time, and indeed, the phenomena themselves migrate over time. At the same time, the application requires prediction and sampling in near real-time. Neither of the traditional approaches for studying spatiotemporal phenomena can adequately deal with both these challenges. Statically deployed sensor networks may not have sensors in the most relevant locations at a given time (and an excessively high sensor density is likely to disturb the phenomenon).

On the other hand, a system consisting purely of mobile sensor nodes tends to be too slow in tracking the phenomenon, in particular over a large area.

We therefore propose a hybrid approach, combining a larger number of static sensors with a few mobile sensing robots. The static sensors will be able to monitor basic attributes in near-real time, and infer potentially interesting undersampled locations. These can then be sampled more densely by the mobile sensors. In addition, the mobile entities will be able to collect samples for offline evaluation by human experts in a laboratory. This allows the system to track attributes for which no autonomous sensing devices have been devised yet.

3 Reactive Observing System

We have constructed a suite of ten sensing/sampling nodes for deployment in natural aquatic ecosystems. The system consists of ten stationary ‘nodes’ that sense pertinent environmental characteristics, collate those measurements into a 2/3-D picture of the ecosystem, and guide a small autonomous surface vehicle to desired sampling locations to retrieve samples. Figure 1 shows (from left to right) the sampling system, the robotic boat, one buoy, and the electronics chassis that mounts inside the buoy. The sampling system is a six-port device custom built for this project. A 36-port version has been designed and is being tested. The prototype boat is a modified RC airboat, equipped with a Garmin GPS and compass for navigation. All modules and sensors have been integrated and connected to the main boat processor board. The main board on both the buoys and the boat is an Intel Stargate. A peripheral basic stamp module is used as the sensor interface. The current sensor suite on each buoy consists of an array of thermistors for sampling temperature at different depths and a fluorometer that can measure the concentration of chlorophyll-a (an indicator of phytoplankton abundance). Communication is based on AODV over an 802.11b wireless connection. Moreover, a land-based weather station will be constructed and integrated into the network to provide pertinent meteorological information for data interpretation.

A sample of the data collected by the experimental setup shown in Figure 1 is depicted in Figure 2. This figure plots the spatial patterns of chlorophyll and temperature in Lake Fulmor at James Reserve, California.

While our current experimental system is on the scale of 10s of stationary nodes, and only one boat, we envisage that as the technology becomes more



Fig. 1. Experimental Setup

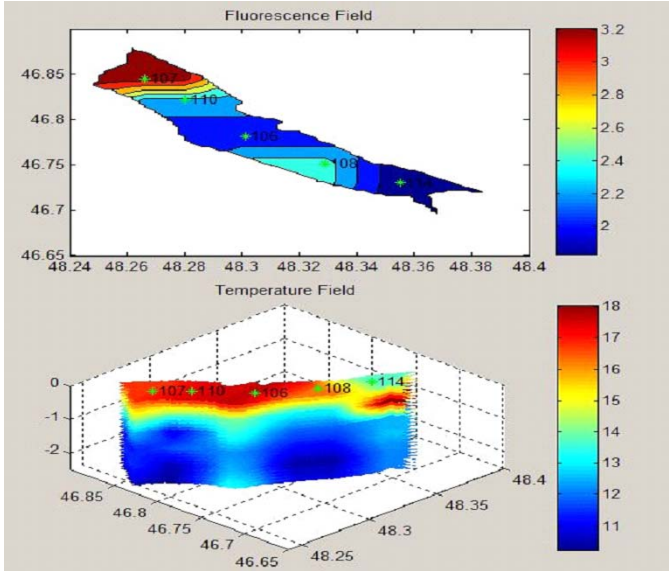


Fig. 2. Empirical Data from Lake Fulmor, James Reserve

commonplace and affordable, the scale of the system will grow significantly, and may eventually monitor large coastal expanses with 100s or 1000s of nodes and 10s of boats. Hence, AMBROSia will be designed to scale gracefully to this size.

Similarly, at the moment, many attributes of interest require analysis of samples in a laboratory, conducted by an expert biologist. However, we anticipate that as analysis technology improves, the system will increasingly be able to determine quantities of interest autonomously. Hence, our system design will allow for the easy inclusion of additional attributes.

4 High-Level View of Our Approach

Figure 3 gives a schematic view of AMBROSia. At the core of AMBROSia is a component for the construction, selection, and adaptation of models. Based on the chosen models, a separate unit optimizes future samples and controls their

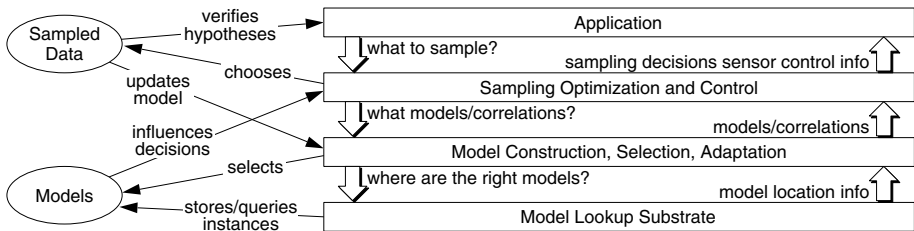


Fig. 3. System Overview

acquisition. The results of these samples in turn affect the decisions of the modeling unit, thus constituting a strong feedback loop in the system. Sampling and control are of course also influenced directly by the requirements of the application, and samples are frequently collected for application-specific purposes, e.g., for scientists to act on. In order to make good decisions about model updates, the model construction unit also requires access to data and fine-grained models stored at individual nodes. In order to find the relevant data, it relies on a model lookup substrate.

Models. At a basic level, a “measurement model” is a representation of a sequence of measurements taken by a sensor. These measurements are sampling some phenomenon, such as temperature or chlorophyll, over a period of time. (Notice that the notion of a model thus extends and subsumes the notion of individual data items and measurements.) Due to resource constraints, the representation will frequently be approximate, and could take the form of a time series, histogram, distribution functions, Hidden Markov Models, or decision tree, for instance.

In principle, measurement models can be drawn from any desired class of models, as the overall system architecture does not prescribe specific models to use. We do note that good models would not only accurately represent the observed data but would also have predictive value and/or be able to extract interesting features from the observed data.

The measurement models are distributed over the physical nodes, which perform and store the measurements. The nodes maintain detailed models. Some forms of these models are reported to other nodes in the system (for analysis). Nodes can make requests for more or less detailed model forms (including full data), based on need and resource availability.

The measurement models are complemented by “external models”. External models are provided by human application experts, and capture prior knowledge about the physics underlying the phenomena. For example, an external model could predict how wind affects water movement between two stationary observation nodes.

Naturally, more powerful inferences and predictions can be made based on combinations of models from multiple sensors, as well as external models. We call such combinations “composite models”. A simple composite model in our example application would be a description of the evolution of the average temperature across a cluster of nodes over time.

Model Lookup. In order to form useful and meaningful composite model, nodes must be able to efficiently locate relevant models at other nodes. Thus, an important component of our research will be the design of a flexible framework for model location, for the purpose of querying or updating them. This lookup system will be akin to distributed naming systems and sensor databases, but due to the higher complexity of our notion of models, will significantly extend

the approaches used in those settings. The design of this component will be integrated into the modeling framework; dynamically instantiated composite models will help guide the lookup operations.

Sampling Optimization and Control. Sophisticated algorithms are needed to control the mobile nodes with changing task assignments in an uncertain and dynamic environment. Moreover, measurement, external, and composite models together allow us to make predictions about future states of the system. These predictions are needed to dynamically control the observation system, and determine the best set of measurements to perform next, subject to resource and time constraints². Thus, another important aspect of the proposed research is the design of algorithms for selecting samples that will likely be of high use to the driving scientific application, or for improving the quality of models. As an illustration, we now give one example formulation of such a problem.

The sampled data at each of the sensor nodes might have varying correlations with each other and with the result of a user query, and hence would make varying degrees of contributions toward resolving the query. In order to minimize processing and communication costs for sensor-network query resolutions, it is crucial to decide how to trade off accuracy in the query-result against sampling a smaller subset of sensor nodes instead. In particular, if at most k sensors can be queried for a given query, what would be the “best” k sensors to choose that would still predict the given query with sufficiently low error, given a priori statistical knowledge about the correlation between the query result and measurements at the sensor nodes?

In its simplest form, this problem can be mathematically formulated as a subset-selection problem in linear regression [7]: Given a set of n random variables X_1, X_2, \dots, X_n (corresponding to measurements at individual sensor nodes) and a predictor random variable Z (corresponding to the query), we are required to select the best k out of the n random variables, such that Z can be predicted by a linear combination of these k random variables with minimum least square prediction error. The only information we are given are the statistical variances and covariances between the X_i and Z variables, obtained from previous data. While this problem has been well known in the statistics community, theoretical progress so far has been limited to greedy and local-search heuristics, without a rigorous analysis of error bounds and time complexities involved [3, 7]. Other variants of this problem have been independently proposed in the mathematics community, such as the sparse approximation problem [8]; theoretical results [5, 8] here have been limited to the special case of nearly orthogonal dictionaries where greedy and convex relaxation methods have been shown to provide $(1 + \epsilon)$ approximation bounds to the optimal solution.

² Newly obtained measurements, along with historic data, are also used to dynamically adjust the models. This direct feedback loop requires that the model validation process be automated and made part of the running system. Thus, an important goal here is to develop a framework for autonomous adaptation of models based on collected measurements.

Summary. The novel aspect of our framework is the strong feedback between the data acquisition process and the modeling process. Traditional sensing systems simply model all attributes of the environment, or a user-specified relevant subset. In contrast, our system will make autonomous data acquisition decisions, which in turn inform the models that guide future decisions. We believe that such a tightly coupled approach will lead to significantly more useful data being collected with the same limited resources.

5 Vision

AMBROSia will aid scientific research by facilitating the testing of scientific hypothesis. It will provide timely predictions of sampling needs. For instance, it might predict that there is a need to increase (in time and/or space) chlorophyll measurements in a particular region in preparation for a possible algae bloom. This prediction might be made based on newly received temperature measurements and wind model predictions. It will also provide tracking information for dynamic phenomena. For instance, it might detect red tide movement and predict better sampling regions for mobile nodes. Overall, our vision for AMBROSia is that it will facilitate observation, detection, and tracking of scientific phenomena that were previous only partially (or not at all) observable and/or understood.

References

- [1] D.M. Anderson. Turning back the harmful red tide. *Nature*, 388:513–514, 1997.
- [2] D.M. Anderson and D.J. Garrison. The ecology and oceanography of harmful algal blooms. *Limnol. Oceanogr.*, 42(5:2):1009–1305, 1997.
- [3] Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams. Maximum-entropy remote sampling. *Discrete Applied Mathematics*, 108:211–226, 2001.
- [4] Pam Belluck. Red tide shuts shellfish areas in new england. *New York Times*, <http://www.nytimes.com/2005/06/04/national/04tide.html>, June 4, 2005.
- [5] A. Gilbert, S. Muthukrishnan, and M. Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, 2003.
- [6] D. J. McGilliguddy, R.P. Signell, C. A Stock, B.A. Keafer, M.D. Keller, R.D. Hetland, and D.M. Anderson. A mechanism for offshore initiation of harmful algal blooms in the coast Gulf of Maine. *Journal of Plankton Research*, 25(9):1131–1138, 2003.
- [7] Alan Miller. *Subset Selection in Regression*. Second edition, 2002.
- [8] Joel Tropp. *Topics in Sparse Approximation*. PhD thesis, The University of Texas at Austin, 2004.

Demonstrating the Validity of a Wildfire DDDAS

Craig C. Douglas^{1,2}, Jonathan D. Beezley⁴, Janice Coen³, Deng Li¹, Wei Li¹,
Alan K. Mandel, Jan Mandel⁴, Guan Qin⁵, and Anthony Vodacek⁶

¹ University of Kentucky, Department of Computer Science, 773 Anderson Hall,
Lexington, KY 40506-0046, USA
{deng.li, wli4}@uky.edu

² Yale University, Department of Computer Science, P.O. Box 208285
New Haven, CT 06520-8285, USA
douglas-craig@cs.yale.edu

³ National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO
80307-3000, USA
janicec@ucar.edu

⁴ University of Colorado at Denver and Health Sciences Center, Department of
Mathematical Sciences, P.O. Box 173364, Denver, CO 80217-3364, USA
{jbeezley, jmandel}@math.cudenver.edu

⁵ Texas A&M University, Institute for Scientific Computation, 612 Blocker, 3404
TAMU, College Station, TX, 77843-3404, USA
guan.qin@tamu.edu.

⁶ Rochester Institute of Technology, Center for Imaging Science, Rochester, NY
14623 USA
vodacek@cis.rit.edu.

Abstract. We report on an ongoing effort to build a Dynamic Data Driven Application System (DDDAS) for short-range forecast of weather and wildfire behavior from real-time weather data, images, and sensor streams. The system changes the forecast as new data is received. We encapsulate the model code and apply an ensemble Kalman filter in time-space with a highly parallel implementation. In this paper, we discuss how we will demonstrate that our system works using a DDDAS testbed approach and data collected from an earlier fire.

1 Introduction

We describe the current state of a dynamic data driven application system (DDDAS) for simulating wildfires. The motivation for this work is the obvious societal value of an accurate forecast compounded with the inherent challenge in modeling this nonlinear, rapidly-changing phenomena, the difficulty in obtaining remote or in situ data about the fire itself, and the challenges of communicating the on-site, out-of-order data of unknown quality to supercomputers and using it to steer the model simulations. The work necessarily extends beyond data assimilation work in progress in atmospheric science due to the specific application

challenges: the model is strongly nonlinear and irreversible, and the data arrives out-of-order from disparate data sources.

The DDDAS is built upon a previously existing coupled atmosphere-wildfire model. Components have been developed and added which (1) save, modify, and restore the state of the atmosphere-wildfire model, (2) apply ensemble data assimilation algorithms to modify ensemble member states by comparing the data with synthetic data of the same kind created from the simulation state, (3) retrieve, process, and ingest data from both novel ground-based sensors and airborne platforms in the near vicinity of a fire, and (4) provide computational results visualized in several ways adaptable to user needs.

2 Summary of the Atmosphere-Fire Model

The original modeling system is composed of two parts: a numerical weather prediction model and a fire behavior model that models the growth of a wildfire in response to weather, fuel conditions, and terrain [1, 2]. These are two-way coupled so that heat and water vapor fluxes from the fire feed back to the atmosphere to produce fire winds, while the atmospheric winds in turn drive the fire propagation. This wildfire simulation model can thus represent the complex interactions between a fire and the atmosphere.

The meteorological model is a three-dimensional non-hydrostatic numerical model based on the Navier-Stokes equations of motion, a thermodynamic equation, and conservation of mass equations using the anelastic approximation. Vertically-stretched terrain-following coordinates allow the user to simulate in detail the airflow over complex terrain. Forecasted changes in the larger-scale atmospheric environment are used to initialize the domain and update lateral boundary conditions. Two-way interactive nested grids capture the outer forcing domain scale of the synoptic-scale environment while allowing the user to telescope down to tens of meters near in the fireline through horizontal and vertical grid refinement. Weather processes such as the production of cloud droplets, rain, and ice are parameterized using standard treatments.

Local fire spread rates depend on the modeled wind components through an application of the Rothermel fire spread formula [3]. The heat release rate is based on [4] which characterizes how the fire consumes fuels of different sizes with time after ignition, distinguishing between rapidly consumed grasses and slowly burned logs. Within each atmospheric grid cell, the land surface is further divided into fuel cells, with fuel characteristics corresponding to the 13 standard fuel types [5]. Four tracers, assigned to each fuel cell, identify burning areas of fuel cells and define the fire front. Fire spread rates are calculated locally along the fire as a function of fuels, wind speed and direction from the atmospheric model (which includes the effects of the fire), and terrain slope, while a local contour advection scheme assures consistency along the fireline. The canopy may be dried and ignited by the surface fire. Then a simple radiation treatment distributes the sensible and latent heat into the lowest atmospheric grid levels.

The empirical fire model is using a submesh representation of the fire region. Within each cell on the fire model grid, the fireline is approximated by a straight line. The fire area in each grid cell is encoded into the position of four points, called tracers. Two of the tracers are generally the intersection of the fireline with the edges of the grid, but not always. This representation makes the fire area hard to adjust in data assimilation (Sec. 3). For this reason, we have developed a translation of the tracers into a level function. The level function is given by values at nodes of the fire grid. The fire region is where the level function is positive. The absolute value of the level function is approximately equal to the Euclidean distance from the fireline. In data assimilation, the level function can be increased or decreased just like the physical quantities in the model.

Our current experiments are with a simple fire model [6], which uses the reaction-convection-diffusion equation for the temperature T and fuel supply S ,

$$c \frac{\partial T}{\partial t} = -\nabla d \nabla T - av \cdot \nabla T + e \frac{\partial S_k}{\partial t} - b(T - T_a), \quad (1)$$

$$\frac{\partial S}{\partial t} = -f(T)S. \quad (2)$$

Eq. (1) is the balance of heat. The term $-\nabla d \nabla T$ models the heat diffusion, $-av \cdot \nabla T$ is the convection by wind with speed v , $e \frac{\partial S_k}{\partial t}$ is the heat generated by burning the fuel, and $-b(T - T_a)$ is the heat lost by flux to the ambient environment with temperature T_a . Eq. (2) is the balance of fuel. The equations are discretized by simple central finite differences in space and the second order trapezoidal method time. The nonlinear system in each timestep is solved by a matrix-free Newton-GMRES method with preconditioning by FFT inversion of the diffusion term. This numerical approach is well suited for a diffusion dominated problem. Methods for convection dominated problems (strong winds relative to the mesh scale) involving upwinding and nonsymmetric preconditioning are in development. This simple model is capable of producing a reasonable fire behavior with an advancing fire front. A more realistic model is under development, which will include several species of fuel, radiative heat transfer between, and evaporation of moisture. It is anticipated that this model will replace the empirical fire model and it will be coupled to the atmospheric model. For related physics based fire models in the literature, see, e.g., [7, 8].

3 How the Ensemble Data Assimilation Works

Ensemble filters work by advancing in time a collection of simulations started from randomly perturbed initial conditions. When the data is injected, the ensemble (called *forecast*) is updated to get a new ensemble (called *analysis*) to achieve a least squares fit using two conditions: change in the ensemble members should be minimized, and the data d should fit the ensemble members state u ,

$$h(u) \approx d, \quad (3)$$

where h is called the *observation function*. The weights in the least squares are obtained from the covariances of the ensemble and of the data error. For

comprehensive surveys of EnKF techniques, see [9, 10, 11]. In general, *EnKF works by forming the analysis ensemble as linear combinations of the forecast ensemble*. This raises two concerns, especially in highly nonlinear models: if the change of state in the update is large, there may not be suitable forecast members to take linear combinations of in order to match the data. Hence, a linear combination of realizable states may not itself be a realizable state. This results in the need for large ensembles and frequent small updates, and has the potential for breakdown.

We are using filters based on the EnKF with data perturbation [12]. But, even with the simple wildfire model (1)-(2), the data assimilation always produces an ensemble with nonphysical solutions and then the simulations breaks down numerically. Therefore, we have proposed a regularization by adding a term involving the change in the spatial gradient of ensemble members to the least squares [13]. Existing ensemble filter formulas assume that the observation function is linear, $h(u) = Hu$, and then compute with the observation matrix H . To simplify the software, we have derived a mathematically equivalent ensemble filter that only needs to evaluate $h(u)$ for each ensemble member. The ensemble update involves computation with large dense matrices. Currently we are using the SCALAPACK parallel linear algebra engine. Future developments include the treatment of nonlinear observation functions, and a hybrid deterministic/Monte Carlo filter that can overcome very strong nonlinearity by modifying the ensemble members to attract them to the truth rather than relying on linear combinations, while maintaining correct ensemble statistics. An approximate one-sided inverse of the observation function is needed for this (Sec. 6). For assimilation of out-of-order data, we will use system states that combine states at several times [6]. The parallel computing framework we have developed was designed with this in mind.

4 Data Transmission Types

Data comes from fixed sensors which measure temperature, radiation, and local weather conditions [14]. The fixed sensors, positioned so as to provide weather conditions near a fire, are mounted at various heights above the ground on a pole with a tripod base. The data logging and transmission electronics are buried in the soil in a protective box. Wiring to the sensors and antennae is insulated. This type of system will survive burn-over by low intensity fires. These sensors supplement other sources of weather data derived from permanent and portable automated weather stations. The temperature and radiation measurements provide the direct indication of the fire front passage and the radiation measurement can also be used to determine the intensity of the fire. The raw data is logged and transmitted as comma delimited ASCII.

Data also comes from images taken by sensors on either satellites or airplanes. The primary source of image data is the Wildfire Airborne Sensor Project (WASP) [15]. This three wavelength digital infrared camera system is carried on an airplane that is flown over the fire area. Camera calibration, an inertial

measurement unit, GPS, and digital elevation data are used in a processing system to convert raw images to a map product with a latitude and longitude associated with each pixel. The three wavelength infrared images can then be processed using a variety of algorithm approaches [15, 16] to extract which pixels contain a signal from fire and to determine the energy radiated by the fire [17, 18]. The original pixel values, the derived probability of fire in each pixel, and the latitude and longitude information are stored as GeoTIFF.

Data from previous fires are stored in a data center in GeoTIFF (images), Excel spreadsheet files, or text files (sensors). The Excel data is made more accessible by converting it to a comma separated value (CSV) format. GPS information is stored about each fixed-location sensor. Each sensor's data is time stamped to identify when the data was collected or received (if it comes without a time stamp). For mobile sensors, both the time stamp and GPS information is available.

Data that comes into the data center must go through a process consisting of up to six steps. *Retrieval*: Get the data from sensors. This may mean receiving data directly from a sensor or indirectly through another computer or storage device (e.g., a disk drive). *Extraction*: The data may be quite messy in raw form, thus the relevant data may have to be extracted from the transmitted information. *Conversion*: The units of the data may not be appropriate for our application. *Quality control*: Bad data should be removed or repaired if possible. Missing data (e.g., in a composite satellite photo) must be repaired. *Store*: The data must be archived to the right medium (or media). This might mean a disk, tape, or computer memory, or no storage device at all (or only briefly) if data is not being archived permanently or only temporarily. *Notification*: If a simulation is using the data as it comes into the data center, the application needs to be informed of the existence of new data.

5 How the Data Moves

In order to run a demonstration simulation, we create a script that lists the sensors that will be included in a run. Multiple runs are handled with a script each.

We are using an Apple XGrid in a University of Kentucky computer science student laboratory to introduce sensor uncertainty into the simulations, independent of the actual scripts we create. The sensors provide data to the running simulation, which is currently in Colorado. The methodology is portable to any Grid environment, not just Apple's.

When a computer in the sensor Grid is idle, data streams regularly based on the scripts we created. We have to do time zone translation on the time stamps when running a simulation in order to be robust. When someone logs into a computer in the sensor Grid and uses the computer, our sensor code is put to sleep automatically by the XGrid controller and the sensors therein are offline until the computer has been idle for some period of time that we cannot control. Using only cycles on idle computers is typical of many Grid environments.

The actual network streaming tool uses a standard TCP client-server scheme. The TCP protocol is a reliable byte stream protocol and underlying network instabilities are accommodated reliably. The programs are small Java codes using built in networking (e.g., write once, write all capabilities) and data encryption methods. The programs are fast and run on almost anything without recompiling.

The sensor network operates either actively (sending data to a specific set of locations) or passively (sending data to requestors). In an active mode, each sensor is represented by a single client (more than one sensor can be on the client, however). When it is time to send data, three way handshaking is used to create a TCP connection to the remote server and send data through it. In a passive mode, each sensor is a server and clients receive data by creating a connection to it and reading data.

The data used by the sensor network is normally kept in a remote data center and only data to be transmitted soon is prefetched. The actual computer load is quite small since much of the time the sensor program is asleep and only periodically does it wake up to do data transfers. What is important is network bandwidth and reliability. Having an acceptable latency time is less important.

6 How the Data Is Injected into the Ensemble Process

The data is related to the model by the observation equation (3). The observation function h maps the system state u to *synthetic data*, which are the values the data would be in the absence of modeling and measurement errors. Knowledge of the observation function, the data, and an estimate of the data error covariance is enough to find the correct linear combinations of ensemble members in the ensemble filter. The data assimilation code also requires an approximate inverse g of the observation function. For a system state u and data d , $g(h(u) - d)$ is the direction in which the system state can change to decrease a norm of the data residual $h(u) - d$. For an observation function that is simply the value of a variable in the system state, the natural choice of approximate inverse can be just the corresponding term of the data residual, embedded in a zero vector.

Building the observation function and its approximate inverse requires conversion of physical units between the model and data, and conversion and interpolation of physical coordinates. In addition, synthetic data at instants of time between the simulation time of ensemble members need to be interpolated to the data time. The data injection itself is done by updating the ensemble to minimize a weighted sum of the data residual and the change in the ensemble (as described in Sec. 3).

The data items enter in a pool maintained by the data acquisition module. The assimilation code can inquire the data acquisition module if there are any new data items available, request their quantitative and numerical properties, and delete them from the pool after they are no longer needed. The properties of the data items include a time stamp, encoding of the type and parameter values of the observation function and its approximate inverse, estimate of the error of the data, and finally the numerical values of the data itself. From the

point of view of the assimilation code, all information about physical units, etc., is encoded in the observation function.

7 How We Deliver Computational Results

Visualization of the model output as an image is accomplished by brightness, color encoding, and transparency for a visual indication of the location and intensity of the fire, and of the probability distribution of the forecast. 3-D visualization of the fire is more complex and complexity increases if high spatial resolution of the output is desired. 3-D visualization uses model output from the fire propagation code for the flame region and from the atmospheric code for visualization of smoke. Ensemble statistics are used for visualization of probability.

The geographic output of the fire model in 2-D or 3-D is visualized in a number of ways. A PDF file of the output as a map is generated for potential output as hardcopy view of the fire at a set point in time. For computer based mapping, manipulation, and visualization of the model output, file formats compatible with the geographic information system (GIS) products are generated.

The time varying output for both 2-D and 3-D is also used to generate a movie playable in any of the media formats, e.g., MPEG. The user may select movie duration up to the maximum extent of the model forecast.

An intuitive and easy method for map visualization is to use a web-based mapping server, e.g., GIS software, Google Maps, or Google Earth. These web-based programs simplify access to map and image data. They let us display model output movies on top of a relevant map background. Within Google Earth, for example, this allows User control of the viewing perspective, zooming into specific sites, and selecting the time frame of the visualization within the parameters of the current available simulation. These web-based programs also allow switching between background types, for example, USGS topographic maps or high resolution satellite images with a road layer or other pertinent layers such water sources added.

8 Conclusions

We are progressing towards a full blown computational test. Data will move, possibly unreliably, from remote sensors to a remote computational machine. Our simulations will be data-driven in terms of the models and scales we use. The choices of models and scales (as in multiscales or resolution) will be made in part based on the data streaming in. We are now in a position to develop the final piece of our DDDAS strategy: having the simulation control how much data is needed and from where in order to improve the quality of the flame wavefront predictions. Only then will we have a truly symbiotic relationship between the running computations and data collection. Our current test should have the right ingredients to predict how our DDDAS will work in a planned future field test with a real wildland fire.

References

1. Clark, T.L., Coen, J., Latham, D.: Description of a coupled atmosphere-fire model. *Intl. J. Wildland Fire* **13** (2004) 49–64
2. Coen, J.L.: Simulation of the Big Elk Fire using using coupled atmosphere-fire modeling. *International J. of Wildland Fire* **14**(1) (2005) 49–59
3. Rothermel, R.C.: A mathematical model for predicting fire spread in wildland fires. USDA Forest Service Research Paper INT-115 (1972)
4. Albini, F.A.: PROGRAM BURNUP: A simulation model of the burning of large woody natural fuels. Final Report on Research Grant INT-92754-GR by U.S.F.S. to Montana State Univ., Mechanical Engineering Dept. (1994)
5. Anderson, H.: Aids to determining fuel models for estimating fire behavior. USDA Forest Service, Intermountain Forest and Range Experiment Station, INT-122 (1982)
6. Mandel, J., Chen, M., Franca, L.P., Johns, C., Puhalskii, A., Coen, J.L., Douglas, C.C., Kremens, R., Vodacek, A., Zhao, W.: A note on dynamic data driven wildfire modeling. In Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J., eds.: *Computational Science - ICCS 2004*. Volume 3038 of *Lecture Notes in Computer Science*. Springer (2004) 725–731
7. Linn, R., Reisner, J., Colman, J.J., Winterkamp, J.: Studying wildfire behavior using FIRETEC. *Int. J. of Wildland Fire* **11** (2002) 233–246
8. Serón, F.J., Gutiérrez, D., Magallón, J., Ferragut, L., Asensio, M.I.: The evolution of a WILDLAND forest FIRE FRONT. *Visual Computer* **21** (2005) 152–169
9. Evensen, G.: The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics* **53** (2003) 343–367
10. Evensen, G.: Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics* (2004) 539–560
11. Tippett, M.K., Anderson, J.L., Bishop, C.H., Hamill, T.M., Whitaker, J.S.: Ensemble square root filters. *Monthly Weather Review* **131** (2003) 1485–1490
12. Burgers, G., van Leeuwen, P.J., Evensen, G.: Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review* **126** (1998) 1719–1724
13. Johns, C.J., Mandel, J.: A two-stage ensemble Kalman filter for smooth data assimilation. *Environmental and Ecological Statistics*. Conference on New Developments of Statistical Analysis in Wildlife, Fisheries, and Ecological Research, Oct 13–16, 2004, Columbia, MI (in print)
14. Kremens, R., Faulring, J., Gallagher, A., Seema, A., Vodacek, A.: Autonomous field-deployable wildland fire sensors. *International J. of Wildland Fire* **12** (2003) 237–244
15. Li, Y., Vodacek, A., Kremens, R.L., Ononye, A., Tang, C.: A hybrid contextual approach to wildland fire detection using multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **43** (2005) 2115–2126
16. Dozier, J.: A method for satellite identification of surface temperature fields of subpixel resolution. *Remote Sens. Environ.* **11** (1981) 221–229
17. Wooster, M.J., Zhukov, B., Oertel, D.: Fire radiative energy for quantitative study of biomass burning: derivation from the BIRD experimental satellite and comparison to MODIS fire products. *Remote Sensing of Environment* **86** (2003) 83–107
18. Smith, A.M.S., Wooster, M., Drake, N., Perry, G., Dipotso, F., Falkowski, M., Hudak, A.: Testing the potential of multi-spectral remote sensing for retrospectively estimating fire severity in African savanna environments. *Remote Sens. Environ.* **97** (2005) 92–115

Development of a Computational Paradigm for Laser Treatment of Cancer

J.T. Oden¹, K.R. Diller², C. Bajaj³, J.C. Browne³, J. Hazle⁴, I. Babuška¹,
J. Bass¹, L. Demkowicz¹, Y. Feng¹, D. Fuentes¹, S. Prudhomme¹,
M.N. Rylander², R.J. Stafford⁴, and Y. Zhang¹

¹ Institute for Computational Engineering and Sciences

² Department of Biomedical Engineering

³ Department of Computer Science,

The University of Texas at Austin, Austin TX 78712, USA

{oden, babuska, bass, leszek, feng, fuentes, serge,
jessica}@ices.utexas.edu,

{kdiller, n.forney}@mail.utexas.edu,

{bajaj, browne}@cs.utexas.edu

⁴ University of Texas M.D. Anderson Cancer Center,

Department of Diagnostic Radiology, Houston TX 77030, USA

{jhazle, jstafford}@mdanderson.org

<http://www.ices.utexas.edu/~feng/dddas>

Abstract. The goal of this project is to develop a dynamic data-driven planning and control system for laser treatment of cancer. The research includes (1) development of a general mathematical framework and a family of mathematical and computational models of bio-heat transfer, tissue damage, and tumor viability, (2) dynamic calibration, verification and validation processes based on laboratory and clinical data and simulated response, and (3) design of effective thermo-therapeutic protocols using model predictions. At the core of the proposed systems is the adaptive-feedback control of mathematical and computational models based on a posteriori estimates of errors in key quantities of interest, and modern Magnetic Resonance Temperature Imaging (MRTI), and diode laser devices to monitor treatment of tumors in laboratory animals. This approach enables an automated systematic model selection process based on acceptance criteria determined a priori. The methodologies to be implemented involve uncertainty quantification methods designed to provide an innovative, data-driven, patient-specific approach to effective cancer treatment.

1 Introduction

Today, cancer is among the leading causes of death in the United States representing 41% of the death rate in 2004 with 1.37 million new cases reported last year. According to the American Cancer Society [1], prostate cancer is responsible for 33% of the cancer deaths of male patients, and breast cancer for 32% of the cancer deaths of female patients in the United States.

Effective cancer treatment requires complete destruction of cancerous cells while maintaining functionality of infected organs. When cancerous tumors occur in well-defined and non-vital regions, conventional surgical procedures for removal of affected tissue is the customary medical treatment; but, traditionally, little can be done when tumors are small and spread over the region. In contrast, laser surgery is minimally invasive and simple to perform, potentially decreasing complications and minimizing hospitalization. Laser therapies provide a lethal dose of heat to the desired site while minimizing damage to surrounding tissue. In particular, laser-induced hyperthermia therapies promise effective treatment of small, poorly-defined metastases or other tumors embedded within vital regions.

The success of laser treatments would be highly increased with the ability to provide a reliable prediction of treatment outcome at the time of treatment delivery using high fidelity computer predictions. Knowledge of temperature history versus time during treatment has been used to predict thermal necrosis in regions where damage is severe, but in regions where temperatures are insufficient to coagulate proteins, the results and subsequent effects have been difficult to predict. This is due, in part, to the expression of heat shock protein (HSP) in the regions of thermal stress, which provide enhanced viability of tumor tissue resulting in the recurrence of cancer. Consequently, knowledge of the thermal dose necessary to activate or de-activate HSP expression as a function of temperature and time in the affected tissue [5, 4] can be critical in planning and implementing an effective thermal treatment by laser surgery.

Grid-computing-enabled dynamic data-driven planning and control systems can provide a unique opportunity for conformal delivery of heat generated by diode or other types of lasers to the target. Image guided thermal ablation therapy surgery or as a complementary therapy for cancer management. In addition, image guidance has the potential to provide real-time treatment monitoring by providing temperature and thermal dose feedback during treatment delivery [7]. By including Magnetic Resonance Temperature Imaging (MRTI), the thermal dose delivered to surrounding normal tissue during therapy can be limited and a more conformal treatment achieved.

2 Description of Laser Treatment Arena

The primary objective of this paper is to describe an approach for guiding laser therapy of cancer, particularly prostate cancer, by accurate control and monitoring of the treatment process through computer simulation. This will be made possible through the development of dynamic data-driven, high-fidelity computer simulation models correlated with *in vivo* spatiotemporal temperature information generated during hyperthermia, and cellular and *in vivo* HSP expression and damage data collected to adaptively control thermo-therapy of cancerous tumors. The specific aims supporting this objective are:

1. To develop an adaptive control system that operates over a computational grid connecting a Treatment/Measurement Arena (TMA) in Houston at

the UT's M.D. Anderson Cancer Center (MDACC) and a Computational/Simulation Arena (CSA) in Austin at The University of Texas at Austin (UT Austin).

2. To develop new algorithms, laboratory and modeling protocols to enable the development of the control systems, including adaptive modeling and meshing procedures, calibration procedures, verification and validation procedures, inverse modeling and sensitivity analysis algorithms, and laboratory procedures for measuring tissue damage and HSP expressions to characterize the kinetic relationship in terms of temperature and time.
3. To demonstrate the effectiveness of the entire process by applying it to the treatment of actual prostate tumors in canines, using modern MRTI-guided laser surgery, distributed visualization and imaging techniques, and data storage and processing devices.

This paper will focus on the mathematical characterization of HSP expression, mesh generation, and development of the computational infrastructure. Development of the control system to operate over a computational grid will be discussed in future work.

3 HSP Characterization and Damage Model

Heat Shock Proteins are a family of gene products expressed in higher concentrations in the presence of environmental stresses. The name vastly understates the HSP family's astounding versatility. These proteins have been identified as critical components of cell survival under adverse environmental

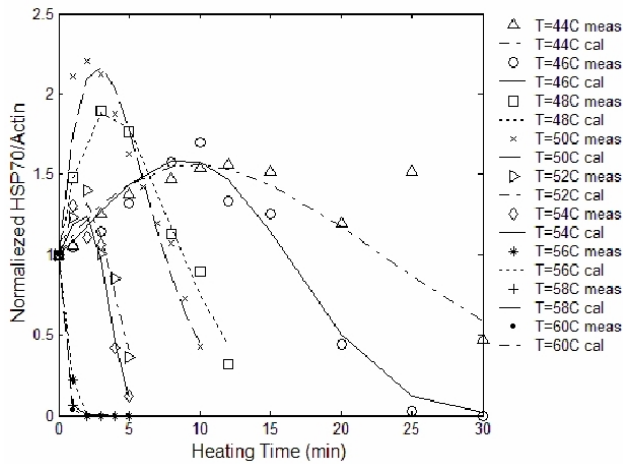


Fig. 1. HSP Expression Model

conditions. Various levels of HSP species indicate the health or likelihood of cell proliferation or drug resistance. Also, measures of cell damage as a function of temperature and time for a specific patient is a critical indication of the effectiveness of thermo-therapies. Knowledge of the temperatures necessary to elicit interaction in tumor resistance and cell damage is essential to effectively produce a desired tissue response and surgical outcome.

The work of Rylander [4] developed a model for HSP expression and cell damage based on an Arrhenius model for a mouse. Figure 1 shows the comparison of the experimentally determined HSP expression with the predicted values of (1). According to Rylander [4], HSP expression $H(T, t)$ can be predicted by:

$$H(T, t) = H_0 e^{\alpha t - \beta t^\gamma} \tag{1}$$

where α , β , and γ are time independent parameters that may depend on temperature, with $\gamma > 1$.

Cellular damage is measured in terms of the damage fraction F_D predicted by means of an Arrhenius integral formulation [5].

$$F_D(t) = 1 - e^{-\Omega(t)} \quad \Omega(t) = \ln(C_0/C_t) = A \int_0^t e^{E_a/RT(\tau)} d\tau$$

where C_0 is the initial concentration of healthy cells, C_t the concentration of healthy cells after heating at time t , A the pre-exponential scaling factor, E_a the activation energy of the injury process, R the universal gas constant, and T the absolute temperature.

4 The Bioheat Transfer Model

Driving the prediction of the HSP and cellular damage is the temperature field produced by the library of bioheat models. The work of Liu [2] has shown Pennes model [3] to give good results for prediction of temperature field in the prostate. As a result only Pennes model will be discussed. Determination of other feasible vasculature and/or continuum bioheat transfer models is in progress.

A space-time variational construction of Pennes equation is used for developing the optimal control framework and in the computational implementation. Following the standard procedure and assuming Cauchy boundary conditions, the variational form of Pennes model is:

Find $T(\mathbf{x}, t) \in L^2([0, \tau], H^1(\Omega))$ with $T'(\mathbf{x}, t) \in L^2([0, \tau], L^2(\Omega))$:

$$B(T; v) = F(v) \quad \forall v \in L^2([0, \tau], H^1(\Omega)) \tag{2}$$

where

$$B(T; v) = \int_0^\tau \int_\Omega \left[\rho c_p \frac{\partial T}{\partial t} v + k \nabla T \cdot \nabla v - \omega c_{blood} (T_a - T) v \right] dx dt$$

$$- \int_0^\tau \int_{\partial\Omega} h T v dA dt + \int_\Omega T(\mathbf{x}, 0) v(\mathbf{x}, 0) dx$$

$$F(v) = \int_0^\tau \int_\Omega Q_{laser}(\mathbf{x}) v dx dt - \int_0^\tau \int_{\partial\Omega} h T_\infty v dA dt + \int_\Omega T^0 v(\mathbf{x}, 0) dx$$

Here k and ω are bounded functions of T , linear in the temperature regimes of interest, c_p and c_{blood} are the specific heats, T_a the arterial temperature, and h is the coefficient of cooling. The laser source term Q_{laser} is a linear function of laser power and exhibits exponential decay with distance from the source. See [5] for additional details.

5 Mesh Generation

The first step in simulation is to construct a finite element mesh used for the governing equations within the library of bioheat transfer models. The mesh generation involves two parts: (1) construction, from MRI data, of a finite element mesh to represent the geometry (2) overlaying the MRTI temperature field onto the finite element mesh.

Prostate tumor cells were inoculated in the hind legs of a mouse and grown to a tumor burden of less than 1.0 cc, Figure 2. A sample of the MRI data used to construct a mesh of the mouse and tumor is shown in Figure 3. The hexahedral mesh of the tumor (blue) and tissue (yellow), shown in Figure 2, was created by a semi-automatic segmentation method adapted to find the interface boundaries of tumor and tissue. Cubic spline and lofting methods are applied to obtain smooth boundaries from the segmented MRI data [5, 8].

Spatio-temporal temperature distribution is measured during the laser treatment with update times less than 5 seconds per image and thickness between planes of 2.0 mm. The temperature field, measured in a canine prostate, is accurately overlaid onto the geometry data and nodal temperature values are assigned to the mesh by taking the interpolant of the MRTI temperature data.

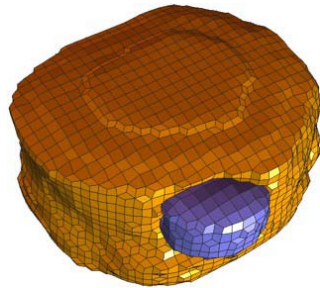


Fig. 2. Mouse and Mesh (from [5])

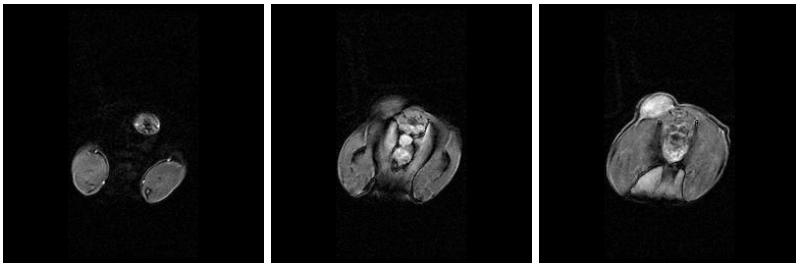


Fig. 3. MRI Data of Mouse

6 Control System

The dynamic flow of *every* problem posed within the laser treatment control system may be cast under the following mathematical framework.

$$\boxed{\begin{aligned} \text{Find } \xi \in \mathcal{W} : Q(\xi) = \inf_{\eta \in \mathcal{M}} Q(\eta) \\ \mathcal{M} = \{\eta \in \mathcal{W} : B(\eta; v) = F(v) \quad v \in \mathcal{V}\} \end{aligned}} \tag{3}$$

where $Q : \mathcal{W} \rightarrow \mathbb{R}$ is a functional characterizing a particular quantity of interest and a saddle point, $(\xi, p) \in \mathcal{W} \times \mathcal{V}$ of the Lagrangian

$$L(\xi; p) = Q(\xi) + F(p) - B(\xi; p)$$

solves the optimal control problem. Thus the problems of calibration, optimization, and computing local computing local error estimates dynamically for each model within the library of bioheat transfer models are to solve:

$$\boxed{\begin{aligned} \text{Find } (\xi, p) \in \mathcal{W} \times \mathcal{V} : \\ B(\xi; v) = F(v) \quad \forall v \in \mathcal{V} \\ B'(\xi; \eta, p) = Q'(\xi, \eta) \quad \forall \eta \in \mathcal{W} \end{aligned}} \tag{4}$$

Algorithms for discretizing and solving the dynamic calibration and optimization problems and dynamically computing local error estimates may be studied through a single example; we consider here the calibration problem. The problem of model calibration is to find the set of model parameters that minimize the L_2 norm of the difference between the predicted temperature field and the experimentally determined temperature field at each time instance of the experimental data. The calibration problem may be stated as follows:

$$\boxed{\begin{aligned} \text{Given experimentally determined temperature fields at times } t_1, t_2, \dots, t_n \\ T_{exp}^{t_1}(\mathbf{x}), T_{exp}^{t_2}(\mathbf{x}), \dots, T_{exp}^{t_n}(\mathbf{x}) \\ \text{Find the temperature field, } T^*, \text{ and best combination of model coefficients, } \\ \beta^* = (k_0^*, k_1^*, \omega_0^*, \omega_1^*), \text{ that minimizes} \\ Q(T, \beta) = \sum_k \int_{\Omega} (T(\mathbf{x}, t_k) - T_{exp}^{t_k}(\mathbf{x}))^2 dx \\ \text{i.e. } (T^*, \beta^*) \text{ satisfies} \\ Q(T^*, \beta^*) = \inf_{(T, \beta) \in \mathcal{M}} Q(T, \beta) \\ \mathcal{M} = \{(T, \beta) \in \mathcal{W} : B(T, \beta; v) = F(v) \quad \forall v \in \mathcal{V}\} \end{aligned}}$$

The semilinear and linear forms are defined from (2) with $k = k(T, k_0, k_1)$ and $\omega = \omega(T, \omega_0, \omega_1)$.

For appropriately chosen test functions, a solution (T, β, p) to the calibration problem satisfies the forward in time state equation with initial conditions,

$$\int_{\Omega} \left[\rho c_p \frac{\partial T}{\partial t} v + k(T, k_0, k_1) \nabla T \cdot \nabla v - \omega(T, \omega_0, \omega_1) c_{blood} (T_a - T) v \right] dxdt - \int_{\partial\Omega} hT v dA dt = \int_{\Omega} Q_{laser}(\mathbf{x}) v dxdt - \int_{\partial\Omega} hT_{\infty} v dA dt \quad \forall v$$

+ initial conditions

a backward in time adjoint/dual equation with terminal conditions ,

$$\int_{\Omega} -\rho c_p \frac{\partial p}{\partial t} \hat{T} + k(T, k_0, k_1) \nabla \hat{T} \cdot \nabla p + \hat{T} \frac{\partial k}{\partial T} \nabla T \cdot \nabla p dxdt + \int_{\Omega} \omega(T, \omega_0, \omega_1) \hat{T} p + \hat{T} \frac{\partial \omega}{\partial T} (T_a - T) p dxdt = 2 \sum_k \int_{\Omega} (T(\mathbf{x}, t_k) - T_{exp}^{t_k}) \hat{T}(\mathbf{x}, t_k) dx \quad \forall \hat{T}$$

+ terminal conditions

and a control equation, which may be shown to be the gradient of the quantity of interest/objective function.

$$\nabla_{\beta} Q = \begin{bmatrix} \int_0^{\tau} \int_{\Omega} \frac{\partial k}{\partial k_0} \nabla T \cdot \nabla p dxdt \\ \int_0^{\tau} \int_{\Omega} \frac{\partial k}{\partial k_1} \nabla T \cdot \nabla p dxdt \\ \int_0^{\tau} \int_{\Omega} \frac{\partial \omega}{\partial \omega_0} (T_a - T) p dxdt \\ \int_0^{\tau} \int_{\Omega} \frac{\partial \omega}{\partial \omega_1} (T_a - T) p dxdt \end{bmatrix} = \mathbf{0}$$

For bioheat models involving larger parameter sets (> 100), a Newton method is used which requires the computation of the Hessian of the objective function. An additional benefit of computing the Hessian is that the Hessian may be used to obtain a stochastic set of model coefficients in which a standard deviation of the computed model coefficients may be obtained. Details of computational algorithms and challenges are to be discussed in a forthcoming paper.

7 Concluding Remarks

This is a truly interdisciplinary project designed to resolve several very challenging research issues. Successful execution of this research requires collaboration and coordination across applied mathematics, computational science, physics, and biomedical engineering. Current progress is showing the mesh generation techniques and computational implementation of Pennes model to give good agreement with experimental MRTI data, Figure 4.

The ability to combine Pennes model with a temperature dependent HSP expression model to be used as a predictive tool [6] for the optimal laser parameters for the laser treatment of cancer is also coming to realization.

Successful completion of this research will provide a powerful methodology for and insight into planning and optimizing the delivery of thermo-therapy for cancer treatments. This study will also enable a better understanding of HSP kinetics and

damage profiles at the cellular and tissue levels in prostate cancer issued from metered thermal stress. The enormous breadth of discipline and the richness of the collaboration required in this effort will set a precedent.

Acknowledgments. The support of this project by the National Science Foundation under grant CNS-0540033 is gratefully acknowledged.

References

1. PROACT, Prostate Action Incorporated. <http://www.prostateaction.org/statistics>. http://www.cancer.org/docroot/STT/stt_0.asp.
2. J. Liu, L. Zhu, and L. Xu. Studies on the three-dimensional temperature transients in the canine prostate during transurethral microwave thermal therapy. *J. Biomech. Engr*, 122:372–378, 2000.
3. H. H. Pennes. Analysis of tissue and arterial blood temperatures in the resting forearm. *J. Appl. Physiol.*, 1:93–122, 1948.
4. M. N. Rylander, Y. Feng, and K. R. Diller. Thermally induced HSP 27, 60, and 70 expression kinetics and cell viability in normal and cancerous prostate cells. *Cell Stress Chaperones*, In review, 2006.
5. M. N. Rylander, Y. Feng, J. Zhang, J. Bass, Stafford R. J., J. Hazle, and K. Diller. Optimizing hsp expression in prostate cancer laser therapy through predictive computational models. *J. Biomed Optics*, In press, 2006.
6. M.N. Rylander. *Design of Hyperthermia Protocols for Inducing Cardiac Protection and Tumor Destruction by Controlling Heat Shock Protein Expression*. PhD thesis, The University of Texas at Austin, 2005.
7. R. Salomir et al. Hyperthermia by MR-guided focused ultrasound: accurate temperature control based on fast MRI and a physical model of local energy deposition and heat conduction. *Magn. Reson. Med.*, 43(3):342–347, 2000.
8. Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195:942–960, 2006.

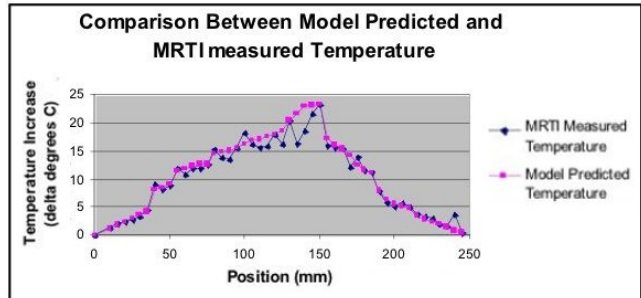


Fig. 4. Model Validation [5]

Blood Flow at Arterial Branches: Complexities to Resolve for the Angioplasty Suite

P.D. Richardson¹, I.V. Pivkin², G.E. Karniadakis², and D.H. Laidlaw³

¹ Division of Engineering

² Division of Applied Mathematics

³ Department of Computer Science

Brown University, Providence, RI 02912, USA

Abstract. This paper presents a case of interdisciplinary collaboration in building and using a set of tools to compute the flows in a branched artery, to compare them with prior physical flow visualization, and to interpret them with further users in mind. The geometry was taken for a typical epicardial coronary artery with a side branch. The incompressible Navier-Stokes equations were solved with the hybrid spectral/hp element solver Nektar. Some simulations were visualized in the CAVE, an immersive 3D stereo display environment, and selected features are described and interpreted.

1 Introduction

Two papers by Colin Caro and his colleagues in the 1970s [1, 2] led to a large increase of interest in the possible role of fluid dynamics in the development of atherosclerosis in arteries, and the consequent impact of this focal and local vessel disease. This disease evolves over time to be critical in chronic conditions such as claudication (from ischemic effects in the legs), transient ischemic effects in the brain (often followed within 6 months or so by thrombotic strokes), and myocardial infarctions due to obstruction in coronary arteries. The local pathology of atherosclerotic plaques is widely found in many arterial locations, in vessel sizes ranging from the aorta down to arteries a tenth or so of the aortic diameter [3, 4]. A pathological finding, that prompted Caro *et al* to suggest a role of local variations in blood flow patterns, is that the distribution of such atherosclerotic lesions on blood vessel walls is not random, but has higher frequency of incidence closely downstream of arterial branches [1, 2]. Clinically the acute and locally-focussed treatment of coronary artery disease is undertaken widely in the angioplasty suite. Cylindrical balloons are introduced to coronary arteries and inflated to open up stenotic sections. Often a physical prop, a stent, is inserted by balloon to hold open a section that had been stenosed (bottlenecked) by the disease. More recently, a catheter-based device has been approved for using a fine wire with a coiled tip to be screwed into a thrombus in the carotid artery and withdraw it, to clear a passage for blood flow there.

Despite many advances made in understanding cell structure and function in the past 30 years, especially regarding transmembrane channels, receptors and

the cytoskeleton, as well as in arterial vessel mechanics and function, including the concept of the vulnerable plaque, on the one hand, and many flow computation and visualization studies in branched and curved tubes on the other hand, no clear closure to the question of why plaques develop spatially non-randomly on arterial walls has been achieved. Even in the flow computations and visualizations, little progress has been made in investigating how well the observations can be explained by combination of prior knowledge of specific flow situations, examined for their breadth by comparison with specific examples. In this paper some selected examinations of the flow in a circular tube in the vicinity of a side branch are described and discussed, first from using regular flow visualization with dye streams and secondly using post-processing of computed flows; the latter embrace tube geometric modifications not achieved in regular physical flow visualization.

2 Physical Flow Visualization for Branched Tubes

Branched tubes were fabricated by machining - drilling, reaming, and chloroform-polishing - of 3mm-diameter straight holes in clear PMMA flat-sided blocks, with 1.5mm-diameter side-branches introduced with axes intersecting those of the primary holes and, in different blocks, having the side branches with their axes at 45 or 90 degrees from the downstream direction of the main 3mm holes. Directly upstream of the blocks, and with their central axes and internal diameter matching the 3mm hole in the block, there were two short tube sections each of which had one small hole permitting a dye stream to be added to the flow, and capable of being rotated so the dye streams from each could be admitted at a selected and measured circumferential angle relative to that of the axis of the side branch downstream. Careful fitting, including use of O-ring seals, assured that the angle of each dye stream could be altered as desired without leaks occurring. Photographs of flow patterns as shown by dye flows were taken with a 35mm Pentax camera having a macro lens.

A typical steady-flow visualization is shown in Figure 1. The flow, passing from left to right, comes very close to the tube junction before showing any alteration from straight laminar flow. In the region where the flow divides the dye streams broaden; because the dye streams have a finite diameter as they approach, this makes clear that the flow develops a different direction close to the wall than at small distances radially inwards. Broadly speaking, one expects the flow close to the wall, having relatively little momentum, to be more sensitive to the local pressure gradient than portions of the flow closer to the center of the stream. The pressure gradient is expected to have components both across the main tube and along it. The transverse component is due to distribution of the reaction force associated with providing the momentum for the portion of the flow going down the side branch. The longitudinal component of pressure gradient is associated with the divergence of streamlines, and the corresponding deceleration of the portion of the fluid stream entering the exit section of the 3mm-diameter tube. This divergence reminds one of a classic solution in fluid dynamics, that of steady

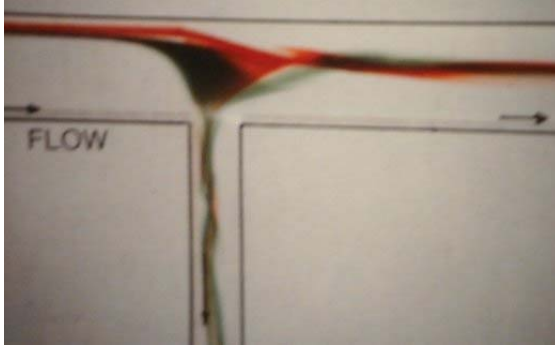


Fig. 1. Steady flow in a tube with a side branch. Visualization illustrating a case with flow separation opposite a side branch.

flow in a conical, diverging tube, and it is interesting that this comparison leads to a prediction of flow conditions - basically a range of values of the side-branch flow rate divided by the downstream main-tube flow rate, as a function of the Reynolds number of the flow upstream of the branch - for which flow separation on the side of the main-tube wall opposite the side-branch, and continuing some distance downstream, is observed. This was reported first as a correlation [5] in 1985 and subsequently with an outline of the analysis [6] at a conference of flows in large blood-vessels held in 1989. Our own computations have confirmed this behavior. Associated with a separated flow there is a significant change in the shear stress in the flow at the wall, and the prospective impact of this on lipid transport between a blood stream and a vessel wall in this region - which happens to be where atherosclerotic plaques are found more frequently - was discussed by Caro *et al* [1, 2].

When the inflow was made pulsatile, the dividing line (between having flow separation or not on the portion of the wall opposite and downstream of the opening into the side branch) broadened into a wedge shape: above the upper line of the wedge the flow was separated there throughout a pulsation, in between the two lines the flow was episodically separated/attached through a pulsation, and below the lower line the flow did not separate at any time during the cycle of a pulsation. The lines came together when the inflow Reynolds number at the upstream portion of the branched tube approached 1,000.

Crossing to the side where the side branch entrance is, the sharp transition of the wall for flow coming along the upstream wall to the side branch leads to a flow separation in the corresponding portion of the entrance section of the side branch over a wider range of flow conditions. This was visible with regular flow visualization, as illustrated in Figures 2(a) and 2(b), where a branch at 45 degrees was examined - this made it easier to photograph the flow from what is the lower-left position in Figure 2(a) to provide Figure 2(b) than would have been the case with a 90-degree branch. Two dyestreams of different colors were applied upstream, and the dye tracks show the dye swerving to enter the side branch in the mid-position across its entrance mouth as seen in Figures 2(a) and 2(b), the

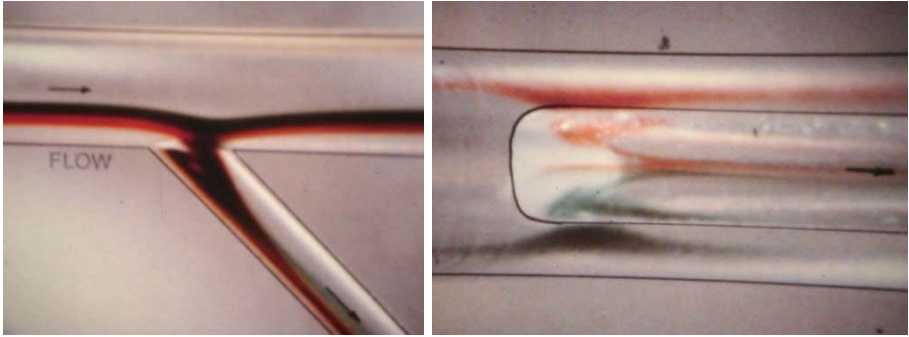


Fig. 2. Steady flow in a tube with a side branch: flow separation in the side branch. (a) Visualization from a side. (b) Visualization from under side branch.

dyestreams approach each other on the underside of the side branch roughly two diameters downstream of the entrance and then each divides, some moving retrograde by the wall and towards the entrance and some continuing downstream. The retrograde flow, on coming very close to the entrance, passes under the flow which has separated at the upstream edge and is carried downstream in the side branch. The length of this region with reversed flow was observed to increase down the side branch as the Reynolds number of the flow in the side branch increased.

3 Visualization of Branched-Tube Flows Derived from Computations

We have made computations for flows, steady and unsteady in branched tubes similar to, and also modified from, those used in the earlier experiments. One modification was to provide curvature to the axes of the tube section, to study more some details of the flow that would be expected in epicardial coronary arteries, which ride in grooves on the curved external surface of the heart. It has been known for several decades following the work of Dean [7] that a secondary motion is found in steady flow in curved tubes.

We thought it would be interesting to examine vortex lines in the flow, and developed some by putting seed points in the flow, computing the local vorticity, marching a small distance in the direction in which it pointed, computing the local vorticity at that point, revising the direction in which to march to the next point, and continued for a few thousand small steps before stopping arbitrarily. If a seed point fell at the edge of the flow field and marching produced a straight line, this was terminated much sooner as an artefact. (In any case, because of the arbitrariness of seed locations and of terminations there would be some lines that appeared to start and finish in the flow, another minor artefact). To visualize the lines they were each sheathed visually with a thin uniform tube, and then displayed as in Figure 3(a). As expected, the central core of the flow showed many similar lines in tight coils traced in each such sequence, with those in

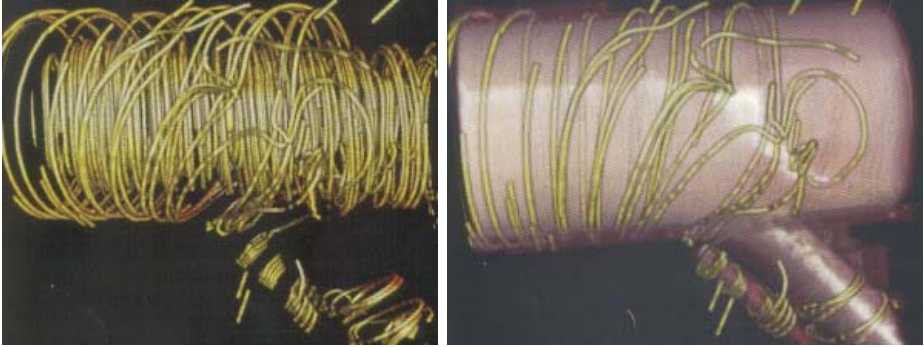


Fig. 3. Steady flow in a tube with a side branch: sample vortex lines from computed flows. (a) Starter tracking points seeded throughout flows in tube. (b) Sheath, of 0.1 mean velocity magnitude surface, inserted to make only vortex lines close to wall distinct.

the outer 10 per cent or so of the radius being inclined differently. To reduce the visual distraction from the inner coiling, an opaque sheath (with some specular reflectivity) was created where the magnitude of the local velocity was 0.1 of the mean velocity in the inflow. A typical resulting image, again close to the side branch, is shown in Figure 3(b). A feature observed here is the formation of 'hairpins' in a few of the vortex lines. These are evocative of similar hairpins that form in laminar boundary layers as part of the process of transition to turbulence, and for any given Reynolds number the likelihood of their being stimulated is increased when there is an 'adverse' pressure gradient, as occurs in this region. This hairpin shape was not observed to evolve downstream, where the pressure gradient stopped being 'adverse', and in this case no early transition to turbulence was indicated. However, it was very interesting to see this, which physical flow visualization was extremely unlikely to catch. It was present in this case only on the side where the vessel wall had negative Gaussian curvature, it was looked for on the opposite side (not shown) but was not present there.

The presence of the reverse flow region at the upstream side of the side branch, under a locally separated flow entering the side branch from the upstream supply, was made visible by retaining the sheathing of regions by local magnitudes of 0.1 mean velocity. Use of the magnitude ignores the local direction of flow. Figure 4 shows a close-up of the region close to the entrance in to a 45-degree branch, and in this Figure the position of the tube walls has not been ghosted in. Visible in this Figure is an object like a finless fish. Because the flow velocity changes direction between its upper surface and the main flow coming into the side branch, there is a seeming gap between them where the local velocity is changing direction and is therefore slow enough to be at less than 0.1 of the entering mean flow. Similarly, to the left of the underside of the 'finless fish' there is a region where the local velocity is less than 0.1, and the location of the tube wall is separated from the visible surface of the 'fish'. With this understanding, the similarity of the phenomenon of a reversed flow region displayed in the images



Fig. 4. Steady flow in a tube with a side branch: surfaces of 0.1 mean velocity magnitude surface shown. Detail near entry to side branch, with 'finless fish' showing part of reverse flow region.

in Figures 2(a) and 2(b) and Figure 4 can be appreciated, in the region just downstream of the lip of the side branch. This is undoubtedly of relevance to tubing geometries in manufactured equipment. Arteries are somewhat rounded locally in this region, and it is likely that the Reynolds number of the approaching flow has to be large enough, or accelerated rapidly enough, for a similar flow separation over a reverse flow region to be present there.

Another feature of the flow in regions affected by a side-branch in tubing geometry is secondary motion. Dean-type secondary motions in curved tubes are already well known, and are driven basically by an imbalance of centrifugal forces in them. The secondary motion involves flow across the diameter of the tube, in the plane of its curvature, and running from the tube wall at the inside of the curve to the outside of the curve, with a compensating flow (to satisfy continuity requirements) close to the wall surfaces that bracket the plane of the curvature. Thus the flow moves around in a "D" shape at each side of the plane of the tube curvature. The secondary flow requires a finite arc of curvature to develop into its full strength: however, this is a fraction of a radian. Even with straight tube sections, the diversion of part of a straight-flowing stream through an angle into a side branch involves a finite period where there can be an imbalance of centrifugal forces across curved streamlines, and a turn of the order of a radian. Flow visualizations (not shown here) suggest that a pair of counter-rotating secondary-flow cores are present for some diameters downstream in a side-branch tube. There may be some in the downstream continuation of the main tube too, especially if there is flow separation opposite the side branch, as the separation region may be occupied by a horse-shoe vortex. Dean-type secondary motions can persist even in the presence of some pulsatility in the flow, and indeed there is some evidence that pulsatility added to a Dean-type secondary motion may increase the mean mass or heat transfer coefficient in a curved tube [8] compared with a straight tube having the same flow through it. Such an increase in mean transfer rates likely involves even larger variations in local transfer coefficients at different circumferential locations. Because the development of atherosclerotic lesions may be

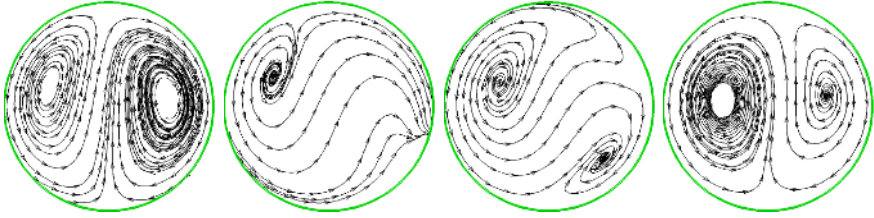


Fig. 5. Secondary flow patterns in main tube from our computations, successively (i) at one tube diameter upstream of intersection of tubing axes (ii) at intersection point of tubing axes (iii) one diameter downstream (iv) three diameters downstream, together showing range of slewing-round of the secondary flow by the flow exiting down the side branch.

affected by local convective lipid transport coefficients, this needs to be examined in greater detail. With space limitations here, we demonstrate one feature relating to secondary flows and a side branch.

In Figure 5 we show a sequence of patterns of secondary flow in the main tube flow, starting one tube-diameter upstream of the point where the axis of the side branch meets the axis of the main tube; at this position the secondary flow is almost identical to that in a curved tube without a side branch. The next secondary flow pattern is at the meeting point of the vessel axes; to the right on this pattern there are streamlines converging towards 3 o'clock, corresponding to contribution to the flow into the side branch, and the vorticity core that had been on the left in the previous flow slewed round roughly 45 degrees clockwise. One diameter further downstream in the main branch the slewing of the vortex structures persists, but with the outflow to the side branch now passed the second vortex core begins to reorganize. By three diameters downstream of the meeting-point of the vessel axes the secondary streamlines are close to normal again. Because this variation of the secondary flow - uniquely made visible by examining the computed flows - will be accompanied by local variations of the mass transport rate at the wall in a way unlikely to be symmetric about the midplane through the parent artery and its side branch, some resolution of the possible role of local transport of lipids may be determined by examining pathological specimens for helical slewing of the atheroma that develop over the range in a parent vessel (such as the left anterior descending coronary artery), downstream of a side branch, in which the secondary flow itself is slewed. Additionally, observations can be made from use of intravascular ultrasound in the angioplasty suite to similar purpose. With pulsatile flows, as was well demonstrable by viewing streamline patterns close to the wall in a CAVE, the secondary flow is strongly sensitive to the accelerating phase in its response.

4 Discussion and Conclusions

Prior experimental work, of which some previously unpublished flow visualization (by the senior author with J. Christo) is included here, posed several

problems about flow in branched tubes and application to human arteries in which atherosclerosis has critical clinical impact. Our studies, of which a segment is described here, have utilized the computational and visualization framework already detailed by Sobel *et al* [9] to examine pulsatile flow in models of branched arteries and, in particular, to search for and examine particular features of interest such as conditions for flow separation opposite the entrance to a side branch, flow separation over the lip of a side branch, local instability, and alteration in secondary flow patterns in a curved tube near a side branch.

In conclusion, this paper illustrates how computational fluid dynamics together with computer visualization form a useful complement to physical experiments and, when broader knowledge of fluid dynamics is also applied, help to make orderly and systematic interpretation of complex phenomena.

Acknowledgements

This work was supported in part by NSF (CNS-0427374, CCR-0086065).

References

1. C. Caro, J.M. Fitz-Gerald, and R.C. Schroter. Arterial wall shear and distribution of early atheroma in man. *Nature*, 223:1159–1161, 1969.
2. C. Caro, J.M. Fitz-Gerald, and R.C. Schroter. Atheroma and wall shear. Observations, correlations, and a proposal for a shear-dependent mass transfer mechanism for atherogenesis. *Proc. Roy. Soc. B*, 177:109–159, 1971.
3. N. Woolf. *Pathology of atherosclerosis*. Butterworth Scientific, London, 1982.
4. M. J. Davies. *Atlas of coronary artery disease*. Lippincott-Raven, Philadelphia PA, 1998.
5. J.L. Christo and P.D. Richardson. Flow in a tube with a side branch. In *Proc 11th Annual North East Bioengineering Conf.*, pages 139–142, 1985.
6. P.D. Richardson and J. Christo. Flow separation opposite a side branch. In D. Liepisch, editor, *Biofluid Mechanics: Blood Flow in Large Vessels*, 1990.
7. W.R. Dean. The streamline motion of a fluid in a curved pipe. *Phil. Mag.*, (7th Ser) 4:208–233, 1927.
8. K. Tanishita, K. Nakano, P.D. Richardson, P.M. Galletti, M. Sugwara, and Y. Sakurai. Augmentation of gas transfer with pulsatile flow in the coiled tube membrane oxygenator design. *Trans. Amer. Soc. Artif. Organs*, 26:561–565, 1980.
9. J.S. Sobel, A.S. Forsberg, D.H. Laidlaw, R.C. Zeleznik, D.F. Keefe, I. Pivkin, G.E. Karniadakis, P. Richardson, and S. Swartz. Particle flurries - synoptic 3d pulsatile flow visualization. *IEEE Comp. Graph. Appl.*, 24(2):76–85, 2004.

A New Architecture for Deriving Dynamic Brain-Machine Interfaces

José Fortes¹, Renato Figueiredo¹, Linda Hermer-Vazquez², José Príncipe¹
and Justin C. Sanchez³

¹Dep. of Electrical and Computer Engineering, ²Dep. of Psychology, ³Dep. of Pediatrics
University of Florida, Gainesville, Florida, USA
{fortes, renato, lindahv, principe, jcs77}@ufl.edu

Abstract. Great potential exists for future Brain Machine Interfaces (BMIs) to help paralyzed patients, and others with motor disabilities, regain (artificial) motor control and autonomy. This paper describes a novel approach towards the development of new design architectures and research test-beds for advanced BMIs. It addresses a critical design challenge in deriving the functional mapping between the subject's movement intent and actuated behavior. Currently, adaptive signal processing techniques are used to correlate neuronal modulation with known movements generated by the subject. However, with patients who are paralyzed, access to the individual's movement is unavailable. Inspired by motor control research, this paper considers a predictive framework for BMI using multiple adaptive models trained with supervised or reinforcement learning in a closed-loop architecture that requires real-time feedback. Here, movement trajectories can be inferred and incrementally updated using instantaneous knowledge of the movement target and the individual's current neuronal activation. In this framework, BMIs require a computing infrastructure capable of selectively executing multiple models on the basis of signals received by and/or provided to the brain in real time. Middleware currently under investigation to provide this data-driven dynamic capability is discussed.

1 Introduction

The seamless integration of brain and body in the healthy human makes us forget that, in some diseases, brains can be deprived of sensory inputs (e.g. vision and audition) or even become isolated from the body (e.g., due to traumatic injury to the spinal cord). These diseases can be mitigated by Brain Machine Interfaces (BMIs) which can be used to deliver inputs from artificial sensors to the appropriate brain cortices and/or use brain signals to directly command and control external robotic devices (see Fig. 1). The focus of this paper is on BMIs for motor control, impressive examples of which include recent demonstrations of monkeys using brain signals captured by electrodes to control computer cursors and robotic arms [1]. Given the brain plasticity, with sensory feedback, external devices may be assimilated and become a part of the user's cognitive space, on a par with body parts. Great potential exists thus for BMIs to help paralyzed patients, and others with motor disabilities, control devices by the intention of movement and regain lost autonomy.

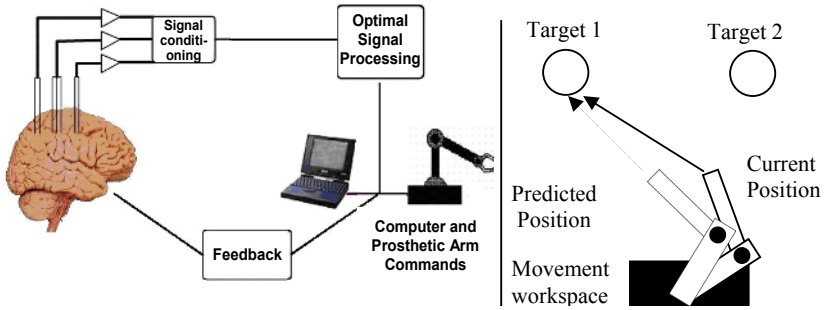


Fig. 1. (Left): BMIs use brain signals to control devices such as computers and robots. Processing is needed to simultaneously execute and adapt the control algorithms as a function of sensed data. (Right): Predictive trajectories based upon neuronal activation and movement kinematics.

The potential of BMIs can only be fulfilled with computer-based systems for not only implementing the decoding algorithms and robotic control but also for improving our understanding of brain function thru modeling. In the specific case of the BMIs, the role of the computer(s) is to dynamically learn and select multiple motor planning and control models to adaptively compute motor control inputs on the basis of brain signals and sensorial feedback. *This paper discusses ideas for facing the challenges of building and using Dynamic Data Driven computing systems to advance the state of the art of research on BMIs.* We use the abbreviation DDBBMI to denote this type of systems.

Extant BMI designs and experiments (e.g. for rodents and monkeys) rely on the subjects' ability to move their limbs (to train the computational models). Through the training of a model (linear adaptive filter or neural network), an optimization criteria such as the mean square error (MSE) is used to minimize the difference between the model output and known movement trajectory (desired response) generated by the subject. In contrast, BMIs that are tested with paralyzed individuals do not have access to the movement trajectory (desired response used for training) because the patient cannot move. One alternative approach to this problem is to incrementally update a *set* of possible movement trajectories using knowledge of the individual's current neuronal activation and the movement target as shown in Fig. 1 (right). Here the goal is to move the robot actuator to one of two targets. However the most appropriate trajectory that is related to the generation of neural activity (intent) is unknown.

We are seeking to model the control loop involving the motor cortex, spinal cord and muscle actuators with greater realism using recent theories of motor control neurophysiology. In this framework, much more intelligence and computing power is required from the assistive device than in present approaches. One must define and choose the set of incremental predictive positions in real-time. Moreover, the set of available trajectories should be seamlessly perceived by the patients. This means that the BMI should operate with low latency and should produce trajectories that are biologically compatible with commands generated by the motor cortex. To avoid instability-causing delays in the sensory feedback loop, the training of the models must be done in a predictive framework using supervised or reinforcement learning.

Due to the demand-based, highly dynamical and distributed nature of brain processing, realistic modeling is nontrivial, involving real-time low-latency computation of many adaptive models that need to be turned on and off throughout the task. Therefore, BMI design and performance will be significantly impacted, in feasibility and capability, by on-demand data-driven adaptation and computation of multiple models of brain learning and motor control. This presents stringent real-time data-driven computational demands which, while potentially deliverable through Grid-computing, require new middleware to be researched and developed. While online DDBMIs are challenging, offline processing is also needed to replay experiments and conduct ancillary modeling studies using “extended” time. They too require significant compute and storage resources, and middleware for dynamically adaptive computation.

2 Envisioned DDBMI Architecture

Pioneering research in motor physiology suggested the existence of *internal models* of motor control. A similar model-based approach to motor control, strongly inspired by the works of Kawato and Wolpert [2][3], is the basis of our design methodology for enhanced BMIs. The first key observation is that a goal-directed movement can be formulated as a sophisticated feedback control problem suggesting that humans visually monitor kinematic events to detect errors in execution (visual cortex), and the parietal cortex appears to be involved in visuomotor aspects of manual manipulative movements. Feedforward control is a skill learned by self-observation of feedback-controlled movements, which involves inverse model learning (e.g. feedback error learning). According to this model, action recognition is mediated by unconscious or implicit mental movement simulation that is implemented in the premotor cortex.

When motor mechanisms are impaired and need to be replaced by assistive technology (e.g. an artificial limb), computational modeling requires particular knowledge of how the cortex may adapt to the new assistive technology. It will be necessary for the different brain areas to relearn how to issue and sequence commands to properly control motion of the assistive device(s). Although the brain plasticity is extraordinary, the assistive device (robotic) interface should act as close as possible to the normal physiology to decrease the training times and preserve as much as possible the accuracy of human movement. We propose to design the interface by specifying the details of the feedforward control loop (in premotor cortex) and its relation with the movement execution in the primary motor cortex. For this purpose, we use a concept called the Multiple Paired Forward-Inverse Model (MPFIM) proposed by Wolpert & Kawato. The MPFIM consists of multiple pairs of models, each comprising a forward model (for movement planning in the premotor cortex) and an inverse model (for movement execution in the primary motor cortex). Individual *model-pairs* or combinations of model-pairs are used to control motion on the basis of real-time feedback data from sensors (visual or proprioceptive).

The left side of Fig. 2 represents in more detail the function of the premotor cortex (forward model, likelihood model and responsibility predictor), where we can recognize the input from prefrontal cortex (efferent copy, and contextual signal), and the visual feedback coming from the parietal cortex. The right side of the figure

contains details of the motor cortex (inverse model) and control scheme. The body moves with its own feedback control loop implemented by an inverse model that receives inputs from the premotor cortex (specifying the desired trajectory) and is self adjusted by the feedback command created by proprioceptive feedback. However, a scheme to select the forward model is still needed. This is accomplished by computing locally the errors of each model (likelihood model) and weighting them across all the models (responsibility predictor). This scheme can be thought as a global weighting gate in a so-called mixture-of-experts paradigm. It is responsible for one of the dynamic data-driven aspects of the DDDBMI architecture. Since each model is a linear filter or a neural network, further adaptation of parameters may be necessary as a result of the input. Our approach uses MPFIM as the basis of DDDBMIs, where, some of the signals of Fig. 2 are generated by the motor, premotor and parietal cortices, and other signals are generated by the robot controller and movement sensors, while the computational components are implemented using Grid computing resources. The premotor and motor cortex electrodes supply the efferent copy and the desired trajectory respectively, while the posterior parietal provides the contextual signals of Fig. 2. Movement sensors (e.g. a camera and object recognition software) will supply the desired trajectory (sensory feedback signals) of Fig. 1.

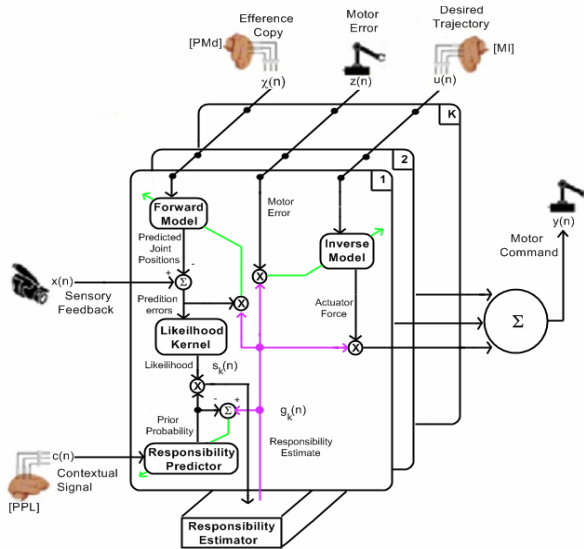


Fig. 2. (Adapted from [2]). The robot controller receives the feedforward motor command and provides the feedback motor command from position sensors

Our approach calls for a *sybiotic simultaneous adaptation* of both the patient motor cortex and the computational models. Since the architecture is anthropomimetic, we conjecture that the patient will be able to learn how to modify his or her brain activity such that it can control the switching of the computational models and ultimately learn the dynamics of the robotic arm. Since the signals for adaptation are being generated on line, we propose to adapt the models in a predictive

framework using either a mixture model or a full blown sequential Bayesian estimation. This will allow the models to track changes in neuronal importance over time. Since the proposed BMI architecture mixes brain signals with robotic control signals, we consider types of models, competitive frameworks and several approaches to the training of individual models that are distinct from those used for other BMIs.

3 Algorithms and Systems Software for Training Mixture Models

We now turn to the question of how to train the gating mechanism (i.e. the individual likelihood models and the responsibility estimator) and forward inverse models of Fig. 2. Let K denote the number of forward models and N denote the number of samples (which define a time series) of the sensory feedback signal $x(n)$ during a given time window. The vector of N samples in the r th window is represented by the vector \mathbf{X}_r . $\chi(n)$ and the corresponding vector $\boldsymbol{\chi}_n$ denote the input to all the models (efferent copy). We choose the free parameters of the predictors (i.e. forward models) and gate that maximize the process log-likelihood.

We consider a system consisting of one hundred paired (inverse controller – forward) linear models, in a mixture-model competitive configuration. The size of this system can be estimated by the following back-of-the-envelope computation. Each model is a MIMO (Multiple Input Multiple Output) system, with one output per coordinate of the output space, and with a number of inputs given by the number of neurons being monitored in the motor cortex. To predict the next position, one needs to use 500 milliseconds of neuronal firing data (which are binned every 50 ms); therefore, the number of parameters is given by ten times the product of the number of neurons by the number of outputs. For 100 neurons this results in $M=3,000$ parameters per linear model, leading to a total of 300,000 parameters total for 100 models. The parameters need to be continuously adapted in real time.

DDDBMI systems for the envisioned applications will have computational demands that exceed the capacity of a single workstation. Grid-computing infrastructures can potentially deliver these necessary resources on demand. However, a DDDBMI system has stringent real-time requirements, as a result of the need for low latency between brain signaling and sensory feedback. We are developing an infrastructure (using In-VIGO¹ Grid middleware) that can aggregate resources, create appropriate parallel execution environments and also guarantee the necessary Quality-of-Service (QoS) in computation and communication to meet response deadlines associated with the above-mentioned latencies. Known parallel implementations of the BMI algorithms will be deployed and managed by this Grid middleware every time a BMI research experiment needs to be done.

The computational component of the proposed DDDBMI architecture is responsible for selectively computing and adapting all the models according to the control flow shown in Fig. 2, and the data signals and algorithms discussed above. The simplified diagram in Fig. 3 shows the broad computational steps associated with

¹ In-VIGO version 1.0 and a prototype portal have been operational and available for public use at <http://invigo.acis.ufl.edu> since 7/2002. A detailed discussion of its design appears in [4].

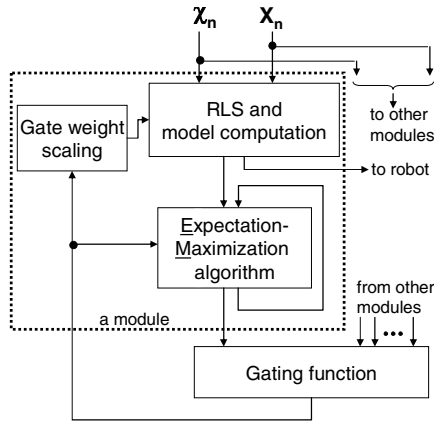


Fig. 3. Procedure for (1) computing and gating models; and (2) adapting weights and contributions of individual models as a function of efferent copy and sensory signals (U_n , χ_n and X_n)

each one of the K sets of models in Fig. 2 when linear models are used. Nonlinear models require additional computation but, for simplicity of exposition, we will not consider them in this part of the paper. As apparent from Fig. 3, with exception of the gating function, all computation is local to a set of models. We will refer to this local computation as a module (it roughly corresponds to an “expert” in the previously-discussed “mixture-of-experts” paradigm). Overall computation starts with the broadcast of U_n , χ_n and X_n to all modules, followed by the independent computation of as many as K modules in parallel. Each module generates two kinds of outputs. One is combined with similar outputs from other modules to generate the three new position coordinates for the robot controller. This output is generated by applying the learned models to the inputs and is hereon referred to as “model computation”. The second kind of output is used by the gating function, which gets similar inputs from other modules in order to determine which modules need to be used to process future inputs, and how the outputs of the modules should be combined to drive the robotic device. The overall computation is dynamically data-driven because the training computation determines through the gating function which modules should be computed and how to weight their outputs.

The model computation and the training computation can be done in parallel for all models. Both occur periodically with new input vectors (corresponding to windows of brain and sensory signal samples) arriving every 50 ms. Each model computation can be computed independently, requires $O(M)$ floating-point multiplications and must be done in less than 50 ms. This is a hard deadline in the sense that live subjects cannot perform in the presence of longer sensory feedback times. Each learning computation can also be computed independently and requires $O(M^2)$ multiplications due to the Recursive Least Squares (RLS) step. The time spent in the training computation and gating function must be less than 100 ms, but this is a soft deadline in the sense that model adaptation can still take place with basis on older inputs. Each module needs to send out the (new and predicted) position coordinates and receives a weight to scale

its outputs. Communication needs are thus modest, estimated as less than 1 ms on a cluster with a 1-Gbps Ethernet. On the basis of our back-of-the-envelope complexity estimates and our experience in implementing single models of similar complexity, an effective computational rate of 100 MFLOPS per module comfortably suffices to do both the model computation and the training computation for a single model with $M=3,000$ parameters with latencies below 50 ms and 100ms, respectively. These rates are well within the reach of optimized implementations of our algorithms on Grid-accessible 3.2GHz-Intel-Xeon machines. The challenge is then for middleware to be able to find and aggregate enough resources to compute and connect all necessary modules.

In-VIGO makes extensive use of virtualization technology for the creation of dynamic pools of virtual resources that can be aggregated on-demand. Its approach to Grid-computing is unique in that it decouples user environments from physical resources by building upon three layers of virtualization. At the bottom layer (*virtual resources*), it manages and aggregates virtual instances of machines, networks, applications and data as needed to build virtual computational Grids to serve specific users and their applications. Within virtual computational Grids, tools and other utilities can be provided as Web services which can be aggregated and combined to constitute (virtual) information Grids. This second layer (*services*) exposes services available to upper middleware levels while hiding the kinds of machines used to provide services. At the topmost layer (*presentation*), In-VIGO users are presented with domain-specific portals (e.g. the nanoHUB nanoelectronics simulation portal) which can be accessed by devices with possibly different interfaces.

Users access the portal through any conventional Web-browser which allows them to log into In-VIGO, request actions and provide data to the User Interface Manager (UIM). The Resource Manager (RM) orchestrates actions of the Virtual Machine System (VMS), Virtual Application System (VAS) and Virtual File System (VFS) that are needed to create the In-VIGO user environment and enable the execution of applications selected by the user. As the names suggest, the VMS, VFS and VAS components provide the virtual instances of machines, file systems and applications needed by a given user's invocation of a tool.

Two broad scenarios are considered for the computational infrastructure needed by DDBMIs. One is the online scenario extensively discussed in this paper, where real-time computation is needed for in-vivo experimentation. The other is the offline scenario where data from past experiments is "replayed" and analyzed, e.g. to generate statistics or train models. In both scenarios, a key requirement is the computation of hundreds of modules as discussed above in relation to Fig. 3. We envision In-VIGO middleware being used to provide BMI researchers with a Web-based interface that would allow them to specify the models they wish to experiment with and possibly additional information (e.g. desired start time, explicit QoS requirements and data collection/storage). In the offline case they would also specify which experiment(s) to replay. Our current research focuses on the challenge of having In-VIGO automatically set up the necessary resources. In the online case this includes setting up the necessary connectivity among resources and the data acquisition system, and guaranteeing QoS requirements. In the offline case, a virtual application will have to be created to re-create experiments from stored data, but data rates can be slowed down, thus removing hard deadlines on computation.

4 Conclusions

Dynamically data-driven BMI's have the potential to drastically impact the ability of patients who suffer from total/partial paralysis following brainstem injury, stroke, or degenerative diseases such as amyotrophic lateral sclerosis (ALS) regain (artificial) motor control and autonomy. We presented our initial ideas for the design of DDDDBMIs and the implementation of a test bed for research of their architectures based on predictive models in a mixture-of-experts configuration inspired by Kawato's work. Our initial analysis of requirements of DDDDBMI's indicates that it is possible to use Grid-computing resources to execute and steer the necessary signal processing tasks on-demand and in real time. We are in the early stages of implementation of the test bed described in this paper.

Acknowledgements

The work reported in this paper is supported in part by the National Science Foundation under NSF Grant No. CNS-0540304.

References

1. Carmena, J.M., et al.: Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biology*, 2003. **1**: p. 1-16.
2. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control, *Neural Networks* 11 (1998) 1317– 1329.
3. Oztop, E., Wolpert, D., Kawato, M.: Mental state inference using visual control parameters *Cognitive Brain Research* 22 (2005) 129– 151, 2004
4. Adabala, S., Chadha, V., Chawla, P., Figueiredo, R., Fortes, J.A.B., Krsul, I., Matsunaga, A., Tsugawa, M., Zhang, J., Zhao, M., Zhu, L., Zhu, X.: "From Virtualized Resources to Virtual Computing Grids: The In-VIGO System", In *Future Generation Computing Systems*, 21(6), April 2005.

Dynamically Adaptive Tracking of Gestures and Facial Expressions*

D. Metaxas, G. Tsechpenakis, Z. Li, Y. Huang, and A. Kanaujia

Center for Computational Biomedicine, Imaging and Modeling (CBIM),
Computer Science Department, Rutgers University,
110 Frelinghuysen Rd, Piscataway, NJ 08854
{dnm, gabrielt, kanaujia}@cs.rutgers.edu, zhli@paul.rutgers.edu,
yuchi.huang@gmail.com

Abstract. We present a dynamic data-driven framework for tracking gestures and facial expressions from monocular sequences. Our system uses two cameras, one for the face and one for the body view for processing in different scales. Specifically, and for the gesture tracking module, we track the hands and the head, obtaining as output the blobs (ellipses) of the ROIs, and we detect the shoulder positions with straight lines. For the facial expressions, we first extract the *2D* facial features, using a fusion between KLT tracker and a modified Active Shape Model, and then we obtain the 3D face mask with fitting a generic model to the extracted *2D* features. The main advantages of our system are (i) the adaptivity, i.e., it is robust to external conditions, e.g., lighting, and independent from the examined individual, and (ii) its computational efficiency, providing us results off- and online with a rates higher than *20fps*.

1 Introduction

Behavioral indicators of deception and behavioral states are extremely difficult for humans to analyze. Our framework aims at analyzing nonverbal behavior on video, by tracking the gestures and facial expressions of an individual that is being interviewed.

Our system uses two cameras (one for the face and one for the whole body view), for analysis in two different scales, and consists of the following modules: (a) head and hands tracking, using Kalman filtering [9] and an data-driven adaptive (to each specific individual) skin regions detection method, (b) shoulders tracking, based on a novel texture-based edge localization method, (c) *2D* facial features tracking, using a fusion between the KLT tracker [12, 15] and different Active Shape Models [2], and (d) *3D* face and facial features tracking, using the *2D* tracking results and our novel *3D* face tracking method. The main advantages of our framework is that we can track both gestures and facial expressions with great accuracy and robustness, in rates higher than *20fps*.

* This research has been funded by an NSF-ITR/NGS-0313134 and an NSF-ITR-[ASE+ECS]-0428231 Collaborative Project to the first author.

This paper is organized as follows. In the next subsection we give a brief overview of the previous work on gestures and face tracking and in section 2 we describe our approach. In subsections 2.1-2.4, we describe the individual parts of our system, namely the head and hands tracking, the shoulders localization, the 2D facial features tracking, and the 3D face tracking, respectively. In section 3 we present the results of our system and in section 4 we present our conclusions.

1.1 Previous Work

Research efforts have investigated gesture analysis [8], but accurate tracking of gestures is still an open topic. According to our work presented in [11], using color analysis, eigenspace-based shape segmentation, and Kalman filters [9], we have been able to track the position, size, and angle of the face and hands regions. In this work we use the color distribution from an image sequence. A Look-Up-Table (LUT) with three color components (red, green, blue) is created based on the color distribution of the face and hands. This three-color LUT, called a 3D-LUT, is built in advance of any analysis and was formed using skin color samples. After extracting the hand and face regions from an image sequence, this method computes elliptical *blobs* identifying candidates for the face and hands. Thus, the most face-like and hand-like regions in a video sequence are identified. Although we have been able to track successfully a wide variety of individuals, we could only use this method under controlled lighting conditions and for a limited range of skin colors.

On the other hand, accurate facial feature localization and tracking under different head poses and facial expressions is a very challenging task. The general problem of feature tracking was presented in the early work of Lucas et.al [12], was developed fully by Tomasi et.al. [15], and was explained explicitly in the paper of Shi et.al. [13]. Later, Tomasi proposed a slight modification which makes the computation symmetric with respect to the two images; the resulting equation is derived in the unpublished note of Birchfield [1].

The appearance of a face can change dramatically as the head poses and the facial expressions change. Sometimes these changes constitute a very difficult problem. Several methods have addressed the issue, such as the multi-view Active Shape Models (ASMs) [2], Active Appearance Models (AAMs) [3], and their extensions [4]. In [10] a summary to multi-view face detection, alignment and recognition is presented, but facial features are not accurately tracked under large head rotations and exaggerated expressions.

Finally, for the 3D face tracking, several algorithms [5, 14] use low-level image feature extraction methods to recover the model state. Most parameterized three dimensional deformable models rely on the correct estimation of image features that are then used to obtain good estimates for the model's state. While the 3D deformable model can reduce some ambiguities to some degree, the unreliable low-level image cues can still cause error accumulation of the 3D model.

2 Our Framework

In this section we describe the different parts of our integrated system. For gesture tracking, we use a camera capturing the entire body of the interviewed individual, whereas a second camera is also used to capture the individual's face in a closer view.

2.1 Head and Hands Tracking

The drawback of our previous approach of [11] is that color is not a robust feature for skin detection and tracking, even if we collect and use in the 3D-LUT a wide variety of skin colors. This method may perform with great accuracy in a controlled environment, i.e., fixed light sources and smooth background (with a color much different from all skin samples of the 3D-LUT), but it is not robust under varying conditions.

To overcome the above problem, we use a data-driven adaptive alternative to the 3D-LUT, i.e., a method that is automatically adapted to the specific individual that is being interviewed. More specifically, our approach consists of the following steps. (i) The first step of our approach is to construct a skin color database to estimate a generic color distribution. Instead of using a 3D-LUT as previously, we estimate a gaussian distribution that fits to the collected color samples. This generic distribution will be modified and adapted to the specific individual's skin color. (ii) In the first frames of the video sequence (optimally 20 frames), we detect the face region of the interviewed individual, using the method proposed by Viola et.al. [16]. This method is experimentally proved to be efficient in terms of computational time, and robust under varying conditions; also it does not require any skin color information and avoids tracking error accumulation over time, since it detects the desired region in each frame separately. (iii) The next step of our approach is to track the face and hands, given the face detection results in the first frames of the sequence. Based on the skin color of the detected facial region, we modify the generic skin distribution (gaussian fitting to the skin color samples, including the specific individual's facial color), and use this new distribution for the detection of the skin regions. In this way, our system automatically adapts to different lighting conditions and all possible skin color variations. (iv) For fast and accurate tracking, free of error drift, we use primarily Kalman tracking [9], and every 10 frames we re-detect the skin regions (face and hands) for the tracker re-initialization. (v) The final result of our method, similarly to our previous work of [11], is the head (face) and hands blobs, extracted with ellipse fitting in the extracted skin regions. To refine our tracking results, we also extract the edges inside the detected skin regions, using the Canny method. For the hand regions, we estimate the edge densities; in cases of individuals with short-sleeve clothes, we segment the hands from the arms based on the hands' increased edge densities (compared to the arms' edge densities). In this way, we extract the blobs of the hand regions and not the entire arms.

2.2 Shoulder Tracking

Tracking the shoulders enables us to detect events such as shrugging, but also estimate relative positions of the hands to the shoulders. In the regions left and right of the estimated head region, we apply the Canny edge detection method. We extract all the edges in each one of these two regions, i.e., the actual (unknown) shoulder edges, background edges, and edges inside the individual's torso region. Our aim is to detect the shoulders, excluding all the undesired edges. To achieve that, we estimate the texture inside these regions, in a block-based manner: we estimate the texture of all blocks centered at the pixels in the normal direction of each edge. In this way, in the normal direction of each edge, at each edge pixel, we obtain a function of texture values. In our system we used the method of Zhang et.al. [17] for the texture estimation. If a detected edge corresponds to the desired shoulder, for all edge points the corresponding texture functions must have a *change-point* at these points respectively. The term *change-point* is obtained from the statistical change-point detection theory [6]; in our framework, in order to detect the texture change points, we use the CUSUM procedure [6]. On the other hand, if an edge does not correspond to a shoulder, the estimated texture functions have either random change-points (not corresponding to the same edge) or no change-points at all. The final result of the shoulder detection is a straight line of fixed length.

2.3 2D Facial Features Tracking

Our framework tracks robustly and accurately facial feature points under multiple views and different expressions. We use an ASM [2] to localize the facial feature points accurately in the first video frame, and to supervise the tracking results in the following frames. For accurate face localization, some local ASMs for the mouth and eyes are used. To track the facial feature points obtained by the multi-pose ASM, we use the KLT tracker [12, 15, 13]. In our framework, this coupling between the KLT tracker and the ASM provides us fast and accurate results, ideal for real-time tracking of facial expressions.

We assume that the face of the interviewed individual is in the frontal view in the first frame of the sequence. We also assume that there is only one person in front of the camera. For the first frame, we use the face detection method of Viola et.al. [16] to obtain a bounding box of the face. Then a mean ASM shape (initialization) is inserted into the bounding box, and the frontal-view ASM is used to localize the facial features accurately. After the facial features initialization, we use the KLT tracker to track the features over time. In parallel to the KLT tracker, we use different ASMs to ensure that the facial features are estimated accurately. This fusion between the KLT tracker and different ASMs makes our framework robust to different head positions and facial expressions.

The frontal-view ASM used in the first frame is trained off-line, using 100 frontal-view face images and their corresponding feature points. For every face we use 87 feature points, which determine different facial features. All these points are manually marked before training. In the ASM, PCA models are trained

both for shape variation and local profile variation. Note that four-level multi-resolution strategy is used here: we train profile variations and search for every point in every level.

Apart from the frontal-view ASM, we also train four ASMs for the left- and right-view, upward and downward head pose, using 70 individuals. These models are used in real-time face tracking to ensure that there is no error accumulation over time that may lead to the loss of track. After the feature points are localized by the frontal-view ASM, the KLT tracker is used to track those feature points over time. In our framework we also integrated Birchfield’s code [1]: for every input frame we use the KLT tracker to track the feature points obtained from the previous frame; then the current points are introduced into one of the ASM shape subspaces. To decide which model will be used for each input frame, we assume that transitions between ASMs are possible only between the frontal-view and one of other four ASMs. We compute the distances between (a) the nose and the left and right cheek, and (b) the nose and the chin. The model is then chosen based on the changes of the ratios of these distances.

2.4 3D Face Tracking

The main advantage of deformable face models is the reduced dimensionality. The smaller number of degree of freedom makes the system more robust and efficient. However, the accuracy and reliability of a deformable model tracking application is strongly dependent on how well the object under tracking fits the family of shapes described by the parameters of the model.

A 3D deformable model is parameterized by a vector of parameters \mathbf{q} . Changes in \mathbf{q} causes geometric deformations of the model. A particular point on the surface is denoted by $\mathbf{x}(\mathbf{q}; \mathbf{u})$ with $\mathbf{u} \in \Omega$. The goal of a shape and motion estimation process is to recover parameter \mathbf{q} from face image sequence. The parameters \mathbf{q} can be divided into two parts: static parameter \mathbf{q}_b , which describes the unchanging features of the face, and dynamic parameter \mathbf{q}_m , which describes the global (rotation and translation of the head) and local deformation (facial expressions) of an observed face during tracking.

The deformations can also be divided into two parts: \mathbf{T}_b for shape and \mathbf{T}_m for motion (expression), so that:

$$\mathbf{x}(\mathbf{q}; \mathbf{u}) = \mathbf{T}_m(\mathbf{q}_m; \mathbf{T}_b(\mathbf{q}_b; s(\mathbf{u}))) \tag{1}$$

The kinematics of the model is $\dot{\mathbf{x}}(\mathbf{u}) = \mathbf{L}(\mathbf{q}; \mathbf{u})\dot{\mathbf{q}}$, Where $\mathbf{L} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}$ is the model Jacobian. Considering the face images under a perspective camera with focal length f , the point $\mathbf{x}(\mathbf{u}) = (x, y, z)^T$ projects to the image point $\mathbf{x}_p(\mathbf{u}) = \frac{f}{z}(x, y)^T$. The kinematics of the new model is given by:

$$\dot{\mathbf{x}}_p(\mathbf{u}) = \frac{\partial \mathbf{x}_p}{\partial \mathbf{x}} \dot{\mathbf{x}}(\mathbf{u}) = \left(\frac{\partial \mathbf{x}_p}{\partial \mathbf{x}} \mathbf{L}(\mathbf{q}; \mathbf{u}) \right) \dot{\mathbf{q}} = \mathbf{L}_p(\mathbf{q}; \mathbf{u}) \dot{\mathbf{q}} \tag{2}$$

Where

$$\frac{\partial \mathbf{x}_p}{\partial \mathbf{x}} = \begin{bmatrix} f/z & 0 & -fx/z^2 \\ 0 & f/z & -fy/z^2 \end{bmatrix} \tag{3}$$

In a physics based deformable model framework, optimization of the parameters is carried out by integrating differential equations derived from the Euler-Lagrange equations of motion:

$$\dot{\mathbf{q}} = \mathbf{f}_q \quad (4)$$

Where the generalized forces \mathbf{f}_q are identified by the displacements between the actual projected model points and the identified corresponding 2D image features. They are computed as:

$$\mathbf{f}_q = \sum_j (\mathbf{L}_p(\mathbf{u}_j))^T \mathbf{f}_{image}(\mathbf{u}_j) \quad (5)$$

Given an adequate model initialization, these forces will align features on the model with image features, thereby determining the object parameters. The dynamic system in equation 4 is solved by integrating over time, using standard differential equation integration techniques:

$$\mathbf{q}(t+1) = \mathbf{q}(t) + \dot{\mathbf{q}}(t)\Delta t \quad (6)$$

Goldenstein et.al. showed in [7] that the image forces \mathbf{f}_{image} and generalized forces \mathbf{f}_q in these equations can be replaced with affine forms that represent probability distributions, and furthermore that with sufficiently many image forces, the generalized force converges to a Gaussian distribution. In our framework, we take advantage of this property by integrating the contributions of ASMs with other cues, so as to achieve robust tracking even when ASM methods and standard 3D deformable model tracking methods provide unreliable results by themselves.

3 Experimental Results

Fig. 1 illustrates an example of tracking the head, hands and shoulders of an individual during an interview, using our framework as described in subsections 2.1 and 2.2. The detected shoulders are shown in red straight lines. The red rectangles illustrate the Kalman tracking results, and the blue rectangles show our tracking results before the blobs estimation, using both Kalman filtering and skin region detection. The head and hands final tracking results (blobs) are shown in white ellipses (along with their major and minor axes).

Fig. 2 illustrates an example of face tracking in six key-frames of a sequence. The the upper images show the results of our 2D face tracking method, whereas the lower images show our results for the 3D face tracking, using the extracted 2D features.

4 Summary and Conclusions

We presented dynamic data-driven framework for tracking gestures and facial expressions from monocular sequences. From the gesture tracking module, we

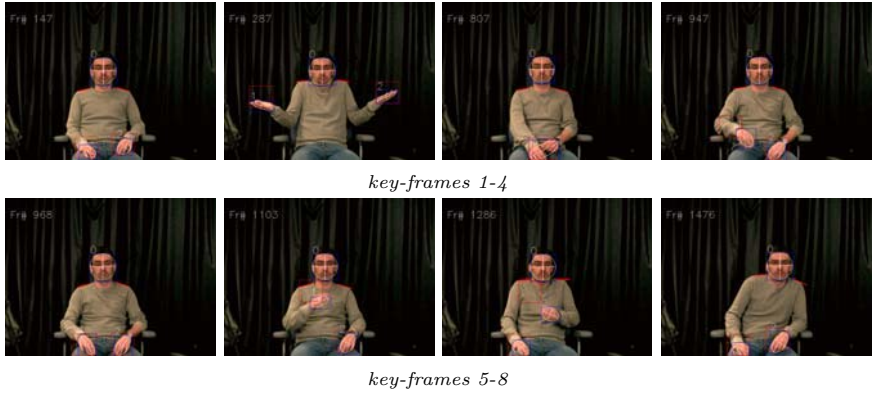


Fig. 1. Head, hands and shoulders tracking results

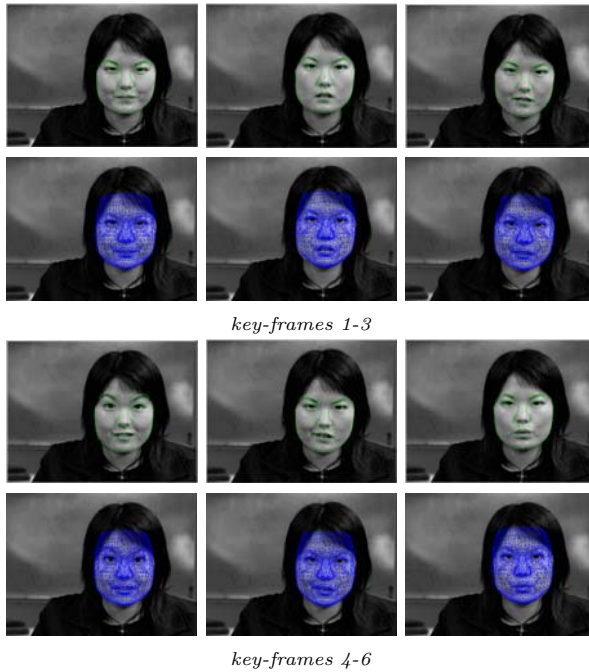


Fig. 2. 2D and 3D face tracking results for six key-frames of a sequence

obtain the blobs (ellipses) of the head and hands, and we detect the shoulder positions with straight lines. For the facial expressions, we first extract the 2D facial features, and then we obtain the 3D face information, using the extracted 2D features. The main advantages of our system are its robustness to lighting changes, its adaptivity to every examined individual, and its computational efficiency, with rates higher than 20 *fps*.

References

1. S. Birchfield, "Derivation of Kanade-Lucas-Tomasi Tracking Equation," *web-published at <http://www.ces.clemson.edu/stb/klf/>*, May 1996.
2. T.F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models - their training and application," *Computer Vision and Image Understanding*, vol. 61(1), p. 389, January 1995.
3. T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Model," *5th European Conference on Computer Vision*, Freiburg, Germany, 1998.
4. T.F.Cootes, and P. Kittipanya, "Comparing Variations on the Active Appearance Model Algorithm," *British Machine Vision Conference*, University of Cardiff, September 2002.
5. D. de Carlo, and D. Metaxas, "Optical Flow Constraints on Deformable Models with Applications to Face Tracking," *International Journal of Computer Vision*, vol. 38(2), pp. 99-127, July 2000.
6. J.L. Devore, *Probability and Statistics for Engineering and the Sciences*, Pacific Grove, Calif.: Brooks/Cole Pub. Co, 2004.
7. S. Goldenstein, C. Vogler, and D. Metaxas, "Statistical Cue Integration in Deformable Models," *Pattern Analysis and Machine Intelligence*, vol. 25(7), pp. 801-813, 2003.
8. D.M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding*, Vol. 73(1), pp.82-98, 1999.
9. R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *ASME Journal of Basic Engineering*, pp. 35-45, March 1960.
10. S.Z. Li, X.L. Zou, Y.X. Hu, Z.Q. Zhang, S.C. Yan, X.H. Peng, L. Huang, and H.J. Zhang, "Real-Time Multi-View Face Detection, Tracking, Pose Estimation, Alignment, and Recognition," *IEEE Conference on Computer Vision and Pattern Recognition, Demo Summary*, Hawaii, December 2001.
11. S. Lu, G. Tsechpenakis, D. Metaxas, M.L. Jensen, and J. Kruse, "Blob Analysis of the Head and Hands: A Method for Deception Detection and Emotional State Identification," *Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 2005.
12. B.D. Lucas, and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *International Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.
13. J. Shi, and C. Tomasi, "Good Features to Track," *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994.
14. H. Tao and T. Huang, "Visual Estimation and Compression of Facial Motion Parameters: Elements of a 3D Model-based Video Coding System," *International Journal of Computer Vision*, vol. 50(2), pp. 111-125, 2002.
15. C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.
16. P.A. Viola, and M.J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57(2), pp. 137-154, 2004.
17. H. Zhang, J.E. Fritts, and S.A. Goldman, "A Fast Texture Feature Extraction Method for Region-based Image Segmentation," *16th Annual Symposium on Image and Video Communication and Processing*, SPIE Vol. 5685, January 2005.

Intelligent Management of Data Driven Simulations to Support Model Building in the Social Sciences

Catriona Kennedy and Georgios Theodoropoulos

School of Computer Science, University of Birmingham, UK
{cmk, gkt}@cs.bham.ac.uk

Abstract. Artificial intelligence (AI) can contribute to the management of a data driven simulation system, in particular with regard to adaptive selection of data and refinement of the model on which the simulation is based. We consider two different classes of intelligent agent that can control a data driven simulation: (a) an autonomous agent using internal simulation to test and refine a model of its environment and (b) an assistant agent managing a data-driven simulation to help humans understand a complex system (assisted model-building). In the first case the agent is situated in its environment and can use its own sensors to explore the data sources. In the second case, the agent has much less independent access to data and may have limited capability to refine the model on which the simulation is based. This is particularly true if the data contains subjective statements about the human view of the world, such as in the social sciences.

For complex systems involving human actors, we propose an architecture in which assistant agents cooperate with autonomous agents to build a more complete and reliable picture of the observed system.

Keywords: agent, cognition, decision support, fault-tolerance, simulation, social sciences.

1 Introduction

In the physical sciences, “dynamic data-driven application simulation” (DDDAS) is a method where data from a physical system is absorbed into a simulation of the system[1]. If DDDAS is applied to an artificial system, the simulation may influence the real physical system (for example, to optimise or adapt it). This is called “symbiotic simulation” because of the mutual benefits of the simulation and the physical system on each other. Examples include semiconductor component testing [2].

The management of a symbiotic simulation has many similarities to a cognitive process. In natural cognitive systems, anticipation is used to direct perception and focus attention on a particular object (e.g. a cup on the table). The reality of the object can modify the further expectancy which may in turn direct attention to further objects or data sources that would not have been anticipated initially (e.g. if the cup is cracked or stuck to the table).

In a symbiotic simulation, predictions can be used as a basis for action on the observed system, just as anticipation does in a cognitive system. If direct action is not appropriate (e.g. because the observed system is not artificial) the predictions can be used to focus on relevant data sources to be assimilated. We will call an agent that controls such a process a “DDDAS agent”.

1.1 Autonomous Agents

An autonomous agent is any robotic or software agent that must continue operation without human intervention. It can use symbiotic simulation to predict states of its environment (or its own components) and to adapt to its environment. We will call this kind of DDDAS agent an “autonomous DDDAS agent” and is a subset of general autonomous agents. Its architecture is shown in Figure 1. Sensors and effectors in the diagram are schematic and may involve software as well as hardware. The predicted state can determine what kind of data is important for subsequent absorption and this requires direction of sensors. Sensor and effector activation can involve a complex translation between high level directions (thick arrows) and the actual low-level measuring or making adjustments multiple thin arrows). Similarly the absorption of data (agent input arrows from sensors) can involve a non-trivial fusion and summarisation process.

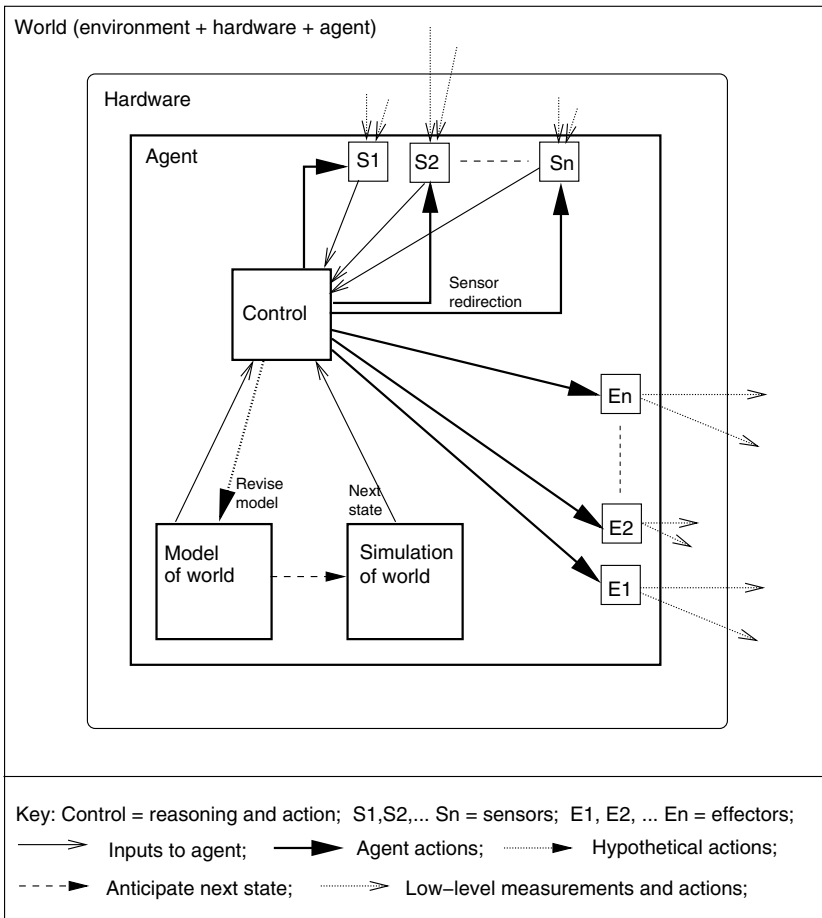


Fig. 1. An autonomous agent using internal simulation and some DDDAS capability

The internal simulation can be just a direct application of the model rules to the current state of the world to predict the next state. Control of sensors for data selection and the subsequent effect on the simulation may be regulated in different ways:

1. Evaluation-directed: if a predicted event is negative (e.g. if a component is predicted to fail), collect more data on the current state of the component; apply more detailed simulations based on the new data;
2. Uncertainty-directed: focus data collection on those areas where there is uncertainty (used in [3] and [4]).
3. Anomaly-directed: E.g. if the sensors indicate much less energy is available than the model predicted then use more sensors to collect data (as sensors may be faulty) and focus the simulation as in (2). If an anomaly persists, it may be the basis for model revision.

Model revision in a fully autonomous agent is ambitious (because of no human intervention). Therefore this action is labelled hypothetical in the diagram.

1.2 Assistant Agents

In contrast to the autonomous agent scenario, Figure 2 shows a scenario in which the DDDAS agent assists with a scientific process. Models and simulations have been developed separately for human understanding (labelled U in the diagram). The purpose of the simulation is primarily to help the scientist or other end-user. The simulation is an external application that the DDDAS agent interacts with.

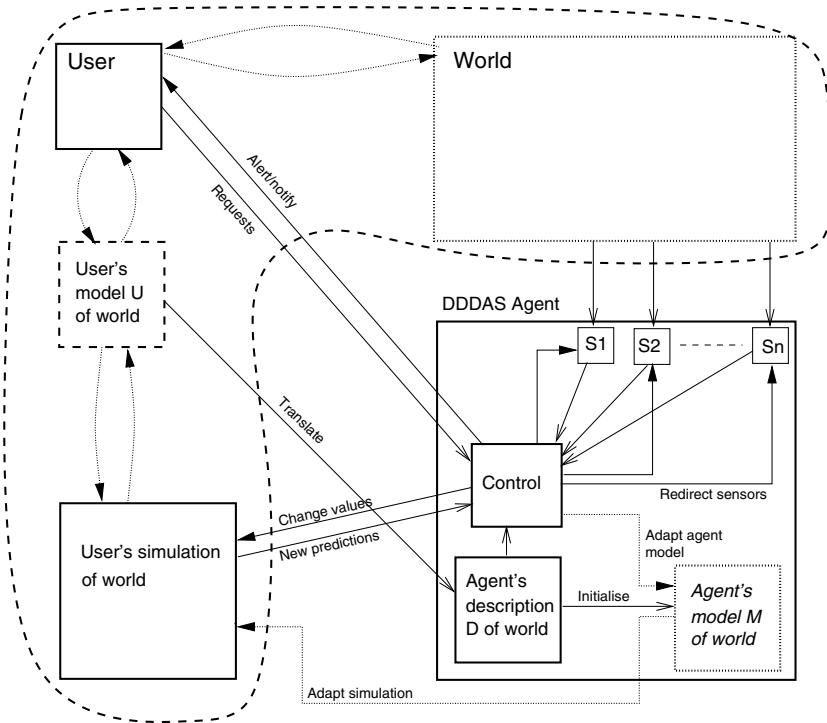
If the agent is to select the relevant data for absorption into the simulation, it must have some description of what is in the simulation and a representation of what the goals and priorities are. The description, which we can call D , can be a representation of the main entities and relations in the original model (U) in a form suitable for agent reasoning (e.g. it could be a set of rules or causal links). It may just cover a subset of U .

The DDDAS agent may also develop its own internal model of the world by adaptation. This is labelled M in the diagram and is an optional enhancement drawn in dotted lines. D can be used to initialise M . Model revision (of U) has to be done by interacting with the human user's understanding of the simulation. M may be used to suggest revisions.

The simulation being managed by the DDDAS agent may include other "agents". They can be natural or artificial systems (e.g. humans, or software). Hence two kinds of "agent" exist: (a) the software DDDAS agent which manages the simulation and (b) the simulated agents, which represent real actors in the observed system. In a social world, examples include individuals, organisations etc. In an artificial world, examples include other software entities.

2 Application of DDDAS to the Social Sciences

Agent-based simulations may be used to model social systems and to assist decision-makers. Existing work includes geographical decision support systems (e.g. [5]) and fire evacuation [6]. A simulation can predict the effects of candidate policies or proposed interventions (or simply the effect of doing nothing). We assume that (a) a set of minimal



Key: Control = reasoning and action of agent; S1,S2,... Sn = sensors controlled by agent;
 ———> Information flow; ———> Agent actions; - - - -> Hypothetical agent actions;
 - - - -> Manual scientific process (experiment, theory formation, simulation).

Fig. 2. A DDDAS agent assisting with modelling and simulation for a human user

requirements have to be met (e.g. relating to environment, health, crime-prevention) and (b) specific needs of participating agents have to be satisfied, with necessity for compromise when conflicts exist. The goals and priorities specified in *D* in Figure 2 are based on these requirements.

Introducing DDDAS into such a simulation leads to the possibility of validating its predictions “online” by continually comparing them with data from the real system. Unexpected features in the observed system can affect the simulation directly and possibly contribute to a revision of the theory that might not have been discovered otherwise.

2.1 How Can DDDAS Be Applied to Social Systems?

For simplicity we assume that a social simulation represents the evolution of a single observed system, not a class of systems. Thus the states of the observed system can fit directly into the simulation states (i.e. agent behaviour in the simulation can be checked against human agent behaviour in the observed system directly). The two systems could

run in parallel (e.g. passengers moving through an airport) or the simulation could be adjusted as historical data becomes available (e.g. housing decisions and mobility in a geographical area). It may also be possible to apply DDDAS to a simulation of a “typical” system, but the data would have to be selected carefully and generalised before being absorbed into the simulation.

The data selection and direction of the simulation can be regulated in the same way as in the physical sciences except that real-time data is not so easily available. The “sensors” will mostly involve database query and data mining tools.

For “evaluation-directed” selection, a negative prediction is a threat to the minimum requirements (e.g. health risk) or an unresolved conflict situation (e.g. leading to violence). (Section 2, (a) and (b)). The DDDAS agent may suggest certain kinds of data that have to be collected (e.g. what kind of behaviours tend to be associated with such events in reality?). It may also redirect the data collection autonomously using interfaces to databases etc. This is expected to be an interactive process.

3 Semantic Grounding

The DDDAS agent may use symbolic reasoning to interpret the simulation states, select appropriate data for absorption and to suggest model revisions. For example, non-monotonic reasoning involves making deductions using tentative assumptions and when necessary making subsequent revisions to these assumptions as new information becomes available (See e.g. [7], Ch. 7). We can apply this to Figure 2 by making updates to M and D interactively.

However, the AI system treats the symbols in D as formal patterns only, and serious errors in the initial version of D may not be detected. This problem has been called “symbol grounding” [8, 9]. We use the broader term “semantic grounding”, to refer to the checking of the validity of a model by independently interacting with the world and if necessary developing new concepts. Interaction with the world does not have to be physical. The important issue is the “data-driven” nature of the concept revision and its potential to fit into a DDDAS architecture.

3.1 Agent Architectures to Support Grounding

Architectures already exist in AI which help to connect symbolic models with independent experience of a physical environment. For example, a hybrid agent architecture such as that in [10] may be appropriate. A hybrid architecture is one that integrates the symbolic tradition of AI with newer behaviour-based approaches such as that introduced by Brookes [11]. Behaviour-based approaches are “data-driven” because the result of the learning process is determined largely by low-level features in the environment and less by any pre-defined knowledge in an ontology. Recent work in concept formation involving symbol grounding includes [12, 13].

An ongoing research question is the degree to which the data-driven layer of a hybrid architecture can “interrupt” or influence the high level reasoning (e.g. if a dangerous situation is detected). Similarly, the degree to which the learning process is data-driven or top-down is important.

3.2 Multiple Ontologies

An alternative method of reducing the brittleness of a symbolic system without actually introducing “grounding” is to use multiple ontologies to introduce fault-tolerance. Multiple ontologies can co-exist as different viewpoints or descriptions of the observed social system. For example, in one description, agents could be modelled with objectively determined states and actions; another might emphasise the beliefs and affective states of agents.

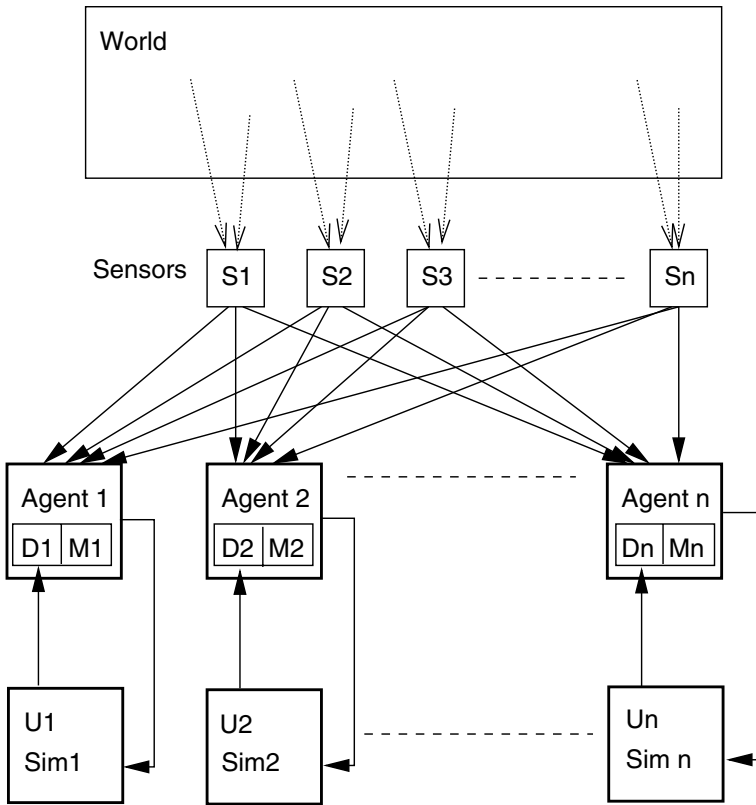


Fig. 3. Multiple agents, each representing a description of the world

Figure 3 shows a configuration involving multiple DDDAS agents, each controlling a simulation Sim_i . Each agent is a copy of the “assistant” type agent introduced earlier in Figure 2 (but with not all arrows and components shown). Each user-defined model U_i is translated into corresponding descriptions D_i for each agent, which may also have its own revisable model M_i . Any major disagreement between agents (and simulations) can be detected and investigated.

4 Integrating Assistant Agents with Autonomous Agents

To implement the hybrid architecture of Section 3.1 in a science assistance scenario, the assistant agent in Figure 2 may be integrated with autonomous agents that can act as “sensing agents”. Similarly, in Figure 3, the “sensors” could have their own autonomous exploration and adaptation capability. However, this would be overridden by high level directives if necessary. Conversely, there may be “alarm” situations where the sensing agent can alert or even interrupt the assistant agent. Although the sensing agents are only semi-autonomous, their architecture can be similar to that of Figure 1 and they may even have their own DDDAS systems.

In most social science scenarios the interaction with the world requires human intermediaries and indirect access via speech acts, meaning that semantic grounding becomes more difficult than in a physical science assistant agent. However, the distinction between the detection of speech acts and the use of sensors is not sharp, since many data sources that could be called “sensors” are actually “event detectors” (i.e. they “say” that an event has occurred). In general, access becomes more indirect (and the concepts less “grounded”) the more the assistant agent relies on information that is pre-processed by another agent and the less control it has over it.

5 Summary and Conclusions

DDDAS has the potential to improve the reliability of simulations used for decision support systems and can also assist with knowledge discovery and creativity in the social science domain. To reduce the brittleness associated with “semantically ungrounded” concepts in social simulations, we can conclude that the following is important:

1. *Autonomous learning and adaptation* is required, involving independent interaction with the world (i.e. data sources) in order to check the validity of a model and to revise it as necessary.
2. *Multiple ontologies* representing alternative descriptions of the world are advantageous. They can be the basis for different models, which can generate their own simulations. Potential problems can be detected if there are significant disagreements between model predictions.
3. *Cooperation between heterogenous agents* acting in different domains and levels of abstraction is important in order to exploit diverse sources of information that can be connected together. This increases the fault-tolerance of the system. The DDDAS agent itself is a service that is embedded in a wider Grid infrastructure and should exchange information with agents managing other services.

Acknowledgements

This research is supported by the Economic and Social Research Council as an e-Social Science feasibility study.

References

1. Darema, F.: Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems. Proceedings of the IEEE: Special Issue on Grid Computing **93** (2005) 692–697
2. Low, M.Y.H., Lye, K.W., Lendermann, P., Turner, S.J., Chim, R.T.W., Leo, S.H.: An Agent-based Approach for Managing Symbiotic Simulation of Semiconductor Assembly and Test Operation. In: Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005), Utrecht, The Netherlands (2005)
3. Plale, B., Gannon, D., Reed, D., Graves, S., Droegemeier, K., Wilhelmson, B., Ramamurthy, M.: Towards Dynamically Adaptive Weather Analysis and Forecasting in LEAD. In: Workshop on Dynamic Data Driven Application Systems at the International Conference on Computational Science (ICCS 2005), Atlanta, USA (2005)
4. Patrikalakis, N., McCarthy, J., Robinson, A., Schmidt, H., Evangelinos, C., Haley, P., Lalis, S., Lermustaux, P., Tian, R., Leslie, W., Cho, W.: Towards a dynamic data driven system for rapid adaptive interdisciplinary ocean forecasting. In Darema, F., ed.: *Dynamic Data-Driven Application Systems*. Kluwer Academic Publishers, Amsterdam (2004)
5. Birkin, M., Dew, P., Macfarland, O., Hodrien, J.: HYDRA: A prototype grid-enabled spatial decision support system. In: First International Conference on e-Social Science, Manchester, UK (2005)
6. Chaturvedi, R., Filatyev, S., Gore, J., Mellema, A.A.: Integrating Fire, Structure and Agent Models. In: Workshop on Dynamic Data Driven Application Systems at the International Conference on Computational Science (ICCS 2005), Atlanta, USA (2005)
7. Rich, E., Knight, K.: Artificial Intelligence. McGraw-Hill Higher Education (1990)
8. Harnad, S.: The symbol grounding problem. *Physica D* **42** (1990) 335–346
9. Edmonds, B., Moss, S.: From KISS to KIDS - an anti-simplistic modelling approach. In: Joint Workshop on Multi-Agent and Multi-Agent-Based Simulation (MAMABS 2004) at the 3rd Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2004), Columbia University, New York City (2004)
10. Sloman, A., Scheutz, M.: A Framework for Comparing Agent Architectures. In: Proceedings of UKCF'02, UK Workshop on Computational Intelligence, Birmingham, UK (2002)
11. Brooks, R.A.: A Robust Layered Control System For A Mobile Robot. *IEEE Journal Of Robotics And Automation* **RA-2** (1986) 14–23
12. Roy, D., Pentland, A.: Learning words from sights and sounds: A computational model. *Cognitive Science* **26** (2002) 113–146
13. Gorniak, P., Roy, D.: Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research* **21** (2004)

Capturing Scientists' Insight for DDDAS

Paul Reynolds, David Brogan, Joseph Carnahan,
Yannick Loitière, and Michael Spiegel

Computer Science Department
University of Virginia

Abstract. One of the intended consequences of utilizing simulations in dynamic, data-driven application systems is that the simulations will adjust to new data as it arrives. These adjustments will be difficult because of the unpredictable nature of the world and because simulations are so carefully tuned to model specific operating conditions. Accommodating new data may require adapting or replacing numerical methods, simulation parameters, or the analytical scientific models from which the simulation is derived. In this research, we emphasize the important role a scientist's insight can play in facilitating the runtime adaptation of a simulation to accurately utilize new data. We present the tools that serve to capture and apply a scientist's insight about opportunities for, and limitations of, simulation adaptation. Additionally, we report on the two ongoing collaborations that serve to guide and evaluate our research.

1 Introduction

In dynamic, data-driven application systems (DDDAS), we have observed that scientists are regularly confronted with the challenge of creating simulations capable of adapting to unanticipated runtime conditions. Runtime conditions may trigger adjustment or replacement of the analytical scientific models from which the system is derived, the numerical methods that implement those models, or the computational infrastructure that executes the numerical methods [1]. How, for example, should a weather simulation respond to newly acquired data that invalidates its predictions? Can some simulation parameters be adjusted automatically by a Kalman filter or must the Kalman filter itself be reparameterized? Perhaps an entirely different underlying model is required to appropriately simulate the new portion of state space exposed by the new data. Because so many aspects of DDDAS are candidates for change, effective automated adaptation to runtime conditions often requires leveraging subject matter expert (SME) insight to guide and constrain the adaptation.

For SME insight to be used in automatic adaptation, the insight must be captured and represented in a way that an automated simulation adaptation system can interpret and understand [2]. To enable this, we are developing formal programming language constructs for describing properties of candidate simulation adaptations. By succinctly encoding a SME's insight about potential simulation adaptations, we seek methods for specifying and automatically exploring simulation expansion opportunities subject to the identified constraints. Extremely

simple examples of such functionality come from the C++ type-conversion procedures where the programmer can specify constraints using typecasts while relegating opportunities for automatic conversion to the compiler. The semantics of the interacting components of DDDAS are much more complex than simple data types. However, we believe it is worthwhile to seek to capture the SME's insight 1) so that SMEs don't have to address particulars early on, 2) so future simulation users can take advantage of early SME insights, 3) so the adaptation process can be conducted in a semi-automated manner, and 4) so future incarnations of a simulation can have a high likelihood of being as efficient as possible.

In this paper, we present ongoing work to encode SME insight for supporting semi-automated program verification and automatic simulation adaptation. Our research is guided by two collaborative simulation development efforts. We report on the role of SME insight for modeling the strong interactions between quark and gluons in hadronic physics simulations and modeling thermoacoustic combustion instabilities in simulations of lean, premixed gas turbine engines.

2 Flexible Points

Opportunities for adapting and fine tuning appear throughout typical simulations because simulations typically include a large number of assumptions with acceptable alternatives. Within simulations themselves we call these opportunities for adaptation *flexible points* [3]. We have addressed a number of issues related to the discovery and use of flexible points, including

- understanding why flexible points work and how they can best be exploited in simulation adaptation,
- studying the relationship between flexible points in a simulation and the simulation's underlying model,
- identifying the limitations of using SME insight to identify assumptions in simulations, and
- mapping out the variety of different flexible points and their possible uses.

To use flexible points more effectively, we have evaluated different simulation optimization techniques for manipulating them. Depending on whether the goal is to develop SME insight about potential adaptations or to exploit SME insight to find the ideal adaptation for a specific problem, different techniques may be more appropriate. Unlike most optimization problems, we are interested in techniques that can be monitored and interrupted by the SME when new insight has been gained or when the optimization appears to be homing in on an unacceptable result [4].

To understand why flexible points work, we have studied a domain we call *coercible software*, which is distinguished by the existence of model abstraction opportunities, where decisions must be made about the level of abstraction to use in simulating phenomena. Because simulations rely heavily on choices of the abstractions for the phenomena that they represent, most simulations are examples of coercible software [5].

To chart the range of flexible points that our language constructs will need to capture, we have developed an evolving taxonomy of flexible points [2, 6]. By distinguishing different types of flexible points, we can develop a toolkit of language constructs that are powerful enough to capture SME insight about a wide variety of simulation adaptations. As we continue to work towards effective tools for capturing and applying SME insight to simulation adaptation, several important challenges remain. These include: analyzing the ways that flexible points can interact with each other, developing requirements for flexible point language constructs, and prototyping and evaluating language constructs for capturing SME insight as flexible points.

To analyze how flexible points interact with each other, we are exploring different assumptions about model abstractions. Starting from the basis established in compositional modeling [7], our goal is to identify those properties that make it possible to automatically manipulate model abstractions without conflicting or unexpected effects. Then, given a set of working assumptions about model abstractions and a taxonomy of flexible points, the requirements for flexible point constructs must be formalized. Lastly, language constructs that meet these requirements must be prototyped and evaluated for use in data-driven simulation adaptation scenarios.

To ensure that our theoretical work on flexible points stays true to the needs of application experts, we have maintained a tightly knit group of researchers who coordinate on end-to-end issues. What good are flexible points if application experts simply require improved visualization tools? We are convinced that flexible points and the technology that supports them are clearly required – a conclusion confirmed by experience gained through our collaborations. In the following sections we describe the application work that is the forge in which our theorizing is tested.

3 Hadronic Physics

When researchers in elementary particle physics utilize traditional methodologies, models are provided in a functional form with free parameters that are adjusted to fit empirical observations. With the acquisition of new data, these free parameters are obvious flexible point candidates for any necessary simulation adaptations. While many methods can optimally tune these flexible points to fit experimental data, there is always some uncertainty as to whether suboptimal tuning results are due to inadequacies in the optimization process or if the model itself is in need of refinements. When data arrives in infrequent batches, the physicist can use domain insight to determine if adjustments to the model are required. In a DDDAS setting, however, data will arrive so frequently that the physicist must have a better understanding of the limitations of the flexible points and the origins of any errors in simulation behavior.

The challenge of identifying and understanding flexible points has become more relevant in recent years because physicists have become increasingly receptive to computational methods derived from the field of artificial intelligence.

Many more tunable parameters are being included in these new simulations and computers are taking a more active role in the very development of the mathematical models. It has become acceptable, for example, to replace quadratics with neural-network-based function approximators [8]. Some argue this is a step forward because the traditional quadratic equations were an artifact of outdated techniques that inserted theoretical bias into the simulations. Others question the utility of neural networks where the voluminous parameters have little intuitive meaning and there are unknown consequences of such underlying biases as the selected network topology and threshold function.

Thus, physicists are concerned with understanding the two types of bias, theoretical and systematic, potentially initiated by the flexible points of their simulations. Theoretical bias is the bias introduced by researchers in the form of the precise structure of the models they use, which invariably constrains the form of the solutions. Systematic bias is the bias introduced by algorithms, such as optimization algorithms, which due to the internal operation of the algorithm may favor some results in ways that are not justified by their objective functions. For example, an optimization algorithm may return a parameterization as its final result without revealing that the returned result is only marginally better than several local minima. The physicist must understand and articulate the impact such behavior, or bias, of the optimization algorithm has on the rest of the simulation.

3.1 The SOMPDF Collaboration: Context and Research Directions

Parton Distribution Functions (PDF) are the distribution of quark and gluon momenta measured during a collision of protons and/or atomic nuclei that are accelerated to relativistic speeds [9]. Finding a functional parameterization of PDFs constitutes a major research effort in elementary particle physics. Physicists design models that provide a quark/gluon distribution, PDF, at a specific energy scale. The simulated PDFs at multiple energy scales are then combined to produce the proton structure functions (observables) that can be matched with the experimental data obtained from supercolliders. To match experimental data well, a global fitting procedure is applied to the parameters of the PDF models. Some physicists have begun to question the global fitting procedure because the χ^2 results of the fit are likely to underestimate both the systematic/theoretical bias and experimental errors from the various data sets. In order to reduce the impact of theoretical assumptions the usage of neural network methods was proposed [8].

To replace the systematic bias injected by a global parameter fitting process, we provide an interactive fitting tool that helps the physicist control the systematic and theoretical bias present in the fitting process, resulting in a PDF model that is better understood. We extend our previous work based on the Self-Organizing Map (SOM) algorithm [6] to create a SOM approach to creating PDFs. Our SOMPDF method is an iterative search process in which the expert interactively delineates the boundary between acceptable and unacceptable results. The SOMPDF method samples the parameter space to generate the results of multiple candidate parameterizations. These results are clustered

into a SOM and judged by the expert. A statistical analysis of the user-selected PDF parameterizations permits the creation of a new set of similar exploratory parameterizations that will be tested in the next iteration of the SOMPDF fitting. This method capitalizes on the strengths of clustering algorithms because the clusters will provide finer-grained statistical distributions than if the data were treated as a monolithic whole. Potentially fruitful pockets of state space can be extracted and explored. Furthermore, the visual properties of SOMs lend themselves particularly well to user interaction. The SOMs are easy to visualize because they are two-dimensional projections of the nonlinear, high-dimensional state space. Additionally, there is a topological ordering over the SOM that ensures similar data from the high-dimensional space will map to nearby datapoints on the two-dimensional SOM. Using intuitive notions of proximity and a simple point-and-click interface, the user will be able to quickly partition the PDF state space.

In this section, we have described how physicists are expanding their PDF simulations to take greater advantage of empirical data and automatic parameterization techniques. The SOMPDF we are developing serves to integrate the physicist's insight with the model parameterization process. Early observations are that the SOMPDF challenges some of the physicist's insights and refines or reinforces others. Not only does the physicist develop new ideas about relevant and related parameters, but the physicist's evaluation functions are adjusted as well. The next section describes another collaboration with SMEs who confront a more challenging case of having to change underlying scientific models.

4 Thermoacoustic Instability

Continuous combustion processes are central to the application of industrial burners, steam and gas turbines, waste generators, and jet and ramjet engines. Under certain conditions, the heat release rate of the combustion process and the dynamic gas pressure of the combustion chamber can become coupled. This coupling will lead to the growth of large-amplitude fluctuations known as thermoacoustic instabilities [10].

Modeling the thermoacoustic problem involves an accurate description of chemical reactions, fluid-dynamics, and acoustic mechanisms of the system. The chemical reactions proceed at time scales of $10^{-2} - 10^{-8}$ seconds, while the other mechanisms proceed at time scales on the order of $10^{-2} - 10^{-4}$ seconds. This is an instance of multiresolution modeling, which can be identified by the need to simulate a unified phenomenon, given several submodels with different levels of temporal or spatial resolution [11]. It remains a hard problem to maintain a consistent representation between different levels of resolution [12]. Two operators are needed to aggregate a set of attributes to a lower level of resolution and to disaggregate an attribute to a higher level of resolution. These operators are often not relatively inverse functions.

A detailed model of chemical kinetics in $CH_4/O_2/N_2$ combustion consists of 17 chemical species in 39 elementary reactions. The detailed model is considered

highly accurate because it matches experiments in studies of gas combustion [13]. But the detailed model is impractical in the study of thermoacoustic coupling. The extremely fast reactions transpire on time scales that are several orders of magnitude higher than the fluid dynamics and pressure acoustics. A one-step kinetic model uses a single global reaction to capture the combustion of methane. The one-step model is an ad hoc kinetic model, meaning the parameters of the model must be tuned to a particular application. The parameters of the model are derived from the Arrhenius form of the reaction rate ($\omega = [CH_4]^\alpha [O_2]^\beta A e^{-\frac{E_a}{RT}}$). The reaction rate (ω) is calculated using four parameters: the CH_4 and O_2 reaction orders (α, β), the pre-exponential factor (A), and the activation energy (E_a). Unfortunately it has been shown that the one-step model is inadequate for simulating the thermoacoustic coupling phenomenon [14].

A one-step model is insufficient because it requires a high activation energy (E_a). The high activation energy causes an amplification of acoustic pressure fluctuations where the detailed model shows a constant magnitude [14]. It is possible that a two-step ad hoc model would behave correctly. In a two-step model, one of the reactions can serve as the rate-limiting reaction. The rate-limiting reaction will prevent the amplification of pressure fluctuations. We are searching the space of two-step models to find an appropriate set of reaction parameters. The two-step model has twice as many degrees of freedom as the one-step model, so there are eight independent parameters. There are many candidate global optimization techniques for minimizing the relative error of the two-step model and detailed model: branch and bound, simulated annealing, stochastic tunneling, genetic algorithms, etc. An additional difficulty is encountered while defining the valid parameter space for the ad hoc two-step model. Most random combinations of the parameter space lead to nonconvergent differential equations in the chemical kinetics simulation [15]. It is impossible to determine a priori which parameter sets do not converge. It is also undecidable whether the chemical kinetics simulation will converge and terminate, given an arbitrary large amount of time.

We have developed a heuristic that searches for converging parameter sets. These parameter sets are subsequently used in the global search techniques for the two-step model. The heuristic begins with an arbitrary pair of (α, β) values and a fixed value for E_a . The space of pre-exponential values is then searched by order-of-magnitude approximation to find those values which quickly converge in the simulation. Once a converging pre-exponential factor is discovered, the space of neighbors is searched to find a value that minimizes the relative error of the models.

We are searching for an appropriate two-step model using the parameter set heuristic. By varying the search domain of (α, β, E_a) different regions of the search space can be studied. In addition there is a set of possible reactions available in the two-step ad hoc model. Whereas the one-step global model is unique, the choice of reactions used in the two-step model is non-unique. We have

not yet found an adequate two-step model. Further analysis is needed to show an existence proof, or a nonexistence proof, for an adequate two-step model.

Often, extensive parameter sweeps are necessary to find satisfactory solutions – in our case converging parameter sets – and are practiced widely by SMEs. In most cases sweeps are employed when SMEs have a high degree of confidence in their models and believe that the ideal solution is certain to be found with the right bindings of parameter values. However, as is often the case, they eventually grow weary of parameter sweeps and begin to explore alternatives. The tension between persisting with sweeps and revisiting model design is often dictated by the depth of tradition associated with a model. In some communities models have existed for decades, and to consider modifying them borders on blasphemy.

In our thermoacoustic coupling work we have learned firsthand that provisions for revisiting assumptions about model abstractions, carried along with a model, would provide an expert with more readily accessible information about available alternatives. Rather than spending months, or years, on parameter sweeps, and then finally turning to question the model itself, an expert could routinely review all of the options available – alternative bindings to decisions made during model design – in a manner consistent with the model designer’s intentions.

5 Conclusion

For the thermoacoustic instability model, extensive parameter sweeps did not reveal a valid set of model parameters for the two-step version of the model. In the end, only the SMEs could say whether any alternatives to the sets of equations employed could even be considered. As a result, no automatic system could explore possible forms for the two-step model. Our experience has demonstrated what the model lacks: design-time capture of flexible point information. On the other hand, the search for a valid model for the Parton distribution functions has benefited considerably from tools that use self-organizing maps to support abstraction and visualization of simulation behavior. With better visualization tools, the SMEs have been able to direct the search for a better parameterizations with far more success than could have been achieved through brute-force parameter sweeps.

Our current belief is that far too much time is spent on parameter sweeps in the application communities. We do not see a clear path to a general approach to DDDAS without resolving this problem. Our investigations of SOMs for parameter identification, and flexible points for formal capture of critical alternatives in selection of model design abstractions, are meant to address the challenges simulationists currently face when seeking to adapt their models to meet desired objectives. We find our work on flexible points and SOMs for parameter identification on the mark for resolving issues that make adaptation slow, and so we are pursuing the ideas aggressively.

Acknowledgments. The authors gratefully acknowledge the support of the NSF under grant 0426971.

References

1. Douglas, C., Deshmukh, A.: Dynamic data-driven application systems: creating a dynamic and symbiotic coupling of application/simulations with measurements/experiments. In: NSF Sponsored Workshop on Dynamic Data Driven Application Systems. (2000)
2. Carnahan, J.C., Reynolds, P.F., Brogan, D.C.: Language constructs for identifying flexible points in coercible simulations. In: Proceedings of the Fall Simulation Interoperability Workshop. (2004)
3. Carnahan, J.C., Reynolds, P.F., Brogan, D.C.: Visualizing coercible simulations. In: Proceedings of the Winter Simulation Conference. (2004) 411–420
4. Waziruddin, S., Brogan, D.C., Reynolds, P.F.: Coercion through optimization: A classification of optimization techniques. In: Proceedings of the Fall Simulation Interoperability Workshop. (2004)
5. Carnahan, J.C., Reynolds, P.F., Brogan, D.C.: Simulation-specific properties and software reuse. In: Proceedings of the Winter Simulation Conference. (2005) 2492–2499
6. Brogan, D.C., Reynolds, P.F., Bartholet, R.G., Carnahan, J.C., Loitière, Y.: Semi-automated simulation transformation for DDDAS. In: International Conference on Computational Science. (2005) 721–728
7. Nayak, P.P.: Causal approximations. *Artificial Intelligence* **70** (1994) 277–334
8. Del Debbio, L., Forte, S., et al.: Unbiased determination of the proton structure function f_p^2 with faithful uncertainty estimation. In: hep-ph/0501067. (2005)
9. Feynman, R.: Photon-Hadron Interactions. W. A. Benjamin, Inc. (1972)
10. Hathout, J.P.: Thermoacoustic instability. In Ghoniem, A.F., ed.: *Fundamentals and Modeling in Combustion*. Volume 2. (1999)
11. Davis, P.K., Bigelow, J.H.: Experiments in multiresolution modeling. RAND Monograph (1998) MR-104.
12. Reynolds, P.F., Srinivasan, S., Natrajan, A.: Consistency maintenance in multiresolution simulation. *ACM Transactions on Modeling and Computer Simulation* **7** (1997) 368–392
13. Peters, N.: Flame calculations with reduced mechanisms an outline. In Peters, N., Rogg, B., eds.: *Reduced kinetic mechanisms for applications in combustion systems*. Volume m15 of *Lecture Notes in Physics*. Springer Verlag (1993) 224240
14. Zambon, A.C.: Modeling of Thermoacoustic Instabilities in Counterflow Flames. PhD thesis, University of Virginia (2005) Department of Mechanical and Aerospace Engineering.
15. Kee, R., Rupley, F., Miller, J.: Chemkin II: A fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. Technical report, Sandia National Laboratories (1989) Sandia Report SAND89-8009.

An MDA-Based Modeling and Design of Service Oriented Architecture

Adel Torkaman Rahmani, Vahid Rafe, Saeed Sedighian, and Amin Abbaspour

Iran University of Science and Technology
Computer Engineering Department
Tehran, Iran

{rahmani, rafe, sedighian, abbaspour}@iust.ac.ir

Abstract. Traditional approaches to software systems development such as using tools and modeling frameworks are appropriate for building individual object oriented or component based software. However they are not suitable for designing of flexible distributed enterprise systems and open environments. In recent years, service-oriented architecture (SOA) has been proposed as a suitable architecture for development of such systems. Most current approaches in employing SOA are tailored to specific domains and hence are not general purpose. Therefore, in order to gain the full benefits of such technology, a more effective general approach to modeling and designing these complex distributed systems is required. In this paper, we present a model-driven approach to SOA modeling and designing complex distributed systems. In this approach, first the PIM of the business system is derived and expressed in standard UML modeling constructs and then this PIM is transformed to the SOA-based PIM by some transforming tool. After the SOA-based PIM is obtained, it can be used to generate PSM for a specific platform such as Web Services, Jini or other platforms. To make it clear how this PSM could be generated we will use Web Services as a target platform and the steps of this transformation will be shown.

1 Introduction

Traditional approaches to development of software systems such as using tools and modeling frameworks are appropriate for building individual object oriented or component-based software. But they are not suitable for designing flexible distributed enterprise systems, the systems that are more complex than before. In fact, the design and implementation of modern software system is complex and costly due to rapid and continuous technology changes. They are complex not only for the volume of code and data involved is much larger but also for many other reasons such as integrating new aspects in those systems (e.g. security, reliability and performance) [1].

In order to find an appropriate solution to development and design of those systems an appropriate paradigm seems necessary. The object-oriented and component-based technology has not significantly met the needs of these systems, and may be considered as adding additional complexity to a domain that needs simplification. And hence a new paradigm like service-oriented architecture [2] is necessary.

SOA is a paradigm that utilizes services as fundamental elements for developing applications [3]. In order to gain the full benefits of such technology, an effective

approach to modeling and designing these complex distributed systems is required. In fact there is not a suitable approach to SOA-based development and little works have been done on this area and most of them are for special applications and specific domains.

To exploit the benefits of SOA effectively and duly, we propose an approach that involves MDA into the context.

This paper presents a model-driven approach to SOA modeling and designing complex distributed systems based on MDA. A new paradigm called Model-Driven Architecture (MDA) has been proposed to develop of complex distributed systems [4]. MDA separates the Platform Independent Model (PIM) from the Platform Specified Model (PSM) of the system and transforming between these models is achieved via appropriate tools.

The paper proposes a new approach to modeling and designing service-oriented architecture. In this approach the PIM of the system is created and then the PSM based on SOA is generated (this PSM is a PIM for next level). Then the final PSM based on a target platform (such as Web Services, Jini and so on) is generated. These models are generated with transformation tools in MDA.

The paper is organized as follows: section 2 explains a brief overview of MDA and SOA. In section 3, an illustrative example is introduced through the related use case diagram. The proposed approach is demonstrated in section 4. In this section the PIM of the MCRI system created with UML will be shown. The steps of transformation from this PIM to PSM based on Web Services will be illustrated. Last but not least, section 5 will conclude the paper.

2 Background

Whereas our research is founded on MDA and SOA approaches, a brief overview of what is aimed by these two approaches is necessary.

2.1 MDA

The OMG's Model Driven Architecture (MDA) separates the modeling task of the implementation details, with out losing the integration of the model and the development in a target platform [5]. The key technologies of MDA are Unified Modeling Language (UML) [6], Meta-Object Facility (MOF), XML Meta-Data Interchange (XMI) [7] and Common Warehouse Metamodel (CWM).

The core paradigm of MDA is model transformation. With MDA, system construction consists of a sequence of models and transformations between these models. The model-driven approach starts with a platform-independent model (PIM), which is subsequently transformed into a platform-specific model (PSM). The PSM is then transformed to code. Generally, PIM and PSM are defined using OMG's Unified Modeling Language (UML) [8].

The main idea of MDA is a pattern: translating from PIM (Platform Independent Model) to PSM (Platform Specific Model) [9]. In theory PIM is a model that is independent from any platform. But for making things actual, we need a final code that works on a specific platform. Using translation process in MDA, we can make some

PSMs for different platforms. Then PSM can be translated to actual code for execution. If in future new platforms or frameworks arise, we can translate our unchanged PIM to these new platforms. MDA pattern can be applied more. In this mechanism that proposed in MDA Guide [10], PSM of Level 1 is PIM for Level 2, etc.

2.2 SOA

SOA exposes real dependencies against artificial ones [11]. A real dependency is a state of affairs in which one system depends on the functionality provided by another. Beside real dependencies there are always artificial dependencies in which the system becomes dependent to configurations and various musts other systems expose. The target of SOA is to minimize artificial dependencies (although it can never be completely removed), and maximize real ones.

This is done via loosely coupling, and the concept of service. A service is a coarse grain functionality objects, with interfaces expressed via a well defined platform independent language. When using services as computational objects, systems can register, find and invoke each other based on a well defined, every one accepted, language hence no one, highly becomes dependent to another system and a high degree of loosely coupling is achieved [12].

3 Illustrative Example

To study the issues of our approach, we have chosen an illustrative example of a Management system which manages some Customer Relationship Information (MCRI).

MCRI system receives queries from customers and sends them the relationship information that is required by the query. Figure 1 presents a simplified use case diagram for a MCRI system.

Figure 1 shows a customer who needs to make a contact. In order to access the contact information, he/she makes a conversation with an employee in the MCRI system. If the customer does not receive his/her requirements then he/she makes a request for appointment and the employee replies the customer's request.

In the next section, after presenting of our proposed architecture, this illustrative example will be modeled and designed based on our proposed approach.

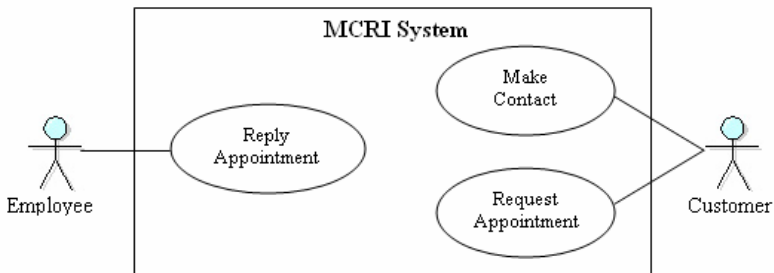


Fig. 1. Use case Diagram of the MCRI System

4 Our Proposed Approach

Modeling service-oriented solutions for above-mentioned systems is not a straightforward task. Due to the complexity of service-oriented solutions (such as defining the scope of services, elements of each service ...) and the diversity of available technology platforms, the MDA approach to designing SOA seems a natural choice. Thus we propose an MDA-based approach, whose framework is shown in figure 2.

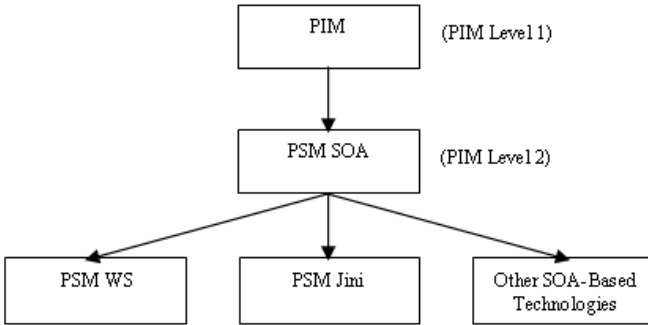


Fig. 2. Proposed Architecture

In this framework, the PIM of the system is created using UML diagrams by the analyst of the system. He/She can analyze the system without worrying about identifying the services and their properties such as scope, loosely-coupling, granularity and so on. Therefore PIM of the system will be designed simply without thinking about services. This is the benefit of the MDA approach.

This phase, designing of PIM, is pretty simple and is accomplished like component-based systems. But what in our proposed approach is important, is how the SOA-based PSM (which is a PIM for the next level) would be derived from the present PIM. The way which is used to identify this PSM must be quite different from the one used to identify PSM in component-based systems; because in component-based systems the patterns which are used to determine the PSM of the system have a specific form. For example, in many of these patterns a ‘set’ and a ‘get’ method is added to the PSM for each attribute in PIM class diagrams. Because of differences between implementing services (in service-oriented systems) and implementing components (in component-based systems), such methods in component based systems are useful but in service oriented systems these methods have a high communication cost. For example, in component-based systems each component is situated in a container which is responsible for managing all the instances of components. In contrast, for each service there is a single instance which manages a set of resources and consequently, unlike components, services are for the most part stateless [13]. This means that we need to view a service as a manager object that can create and manage instances of a type, or set of types. This yields a design pattern that makes use of value objects which represent the instance states, for systems that need

to maintain and use these instant states. This method of passing the state of a service to its client implies that rather than using a large number of small operations to retrieve the service state, only a single large operation is required. This has enormous impacts on network usage for remote services.

According to above discussion, in our approach after creating the PIM, this PIM is transformed -with a transformation tool- to another PIM based on SOA. In fact this second PIM is a PSM.

In this transformation, for each class diagram in PIM, a Service Manager is created that manages the Instant Services. This management involves creation, deletion, updating a service and state management of services. To complete this transformation, we need some other special patterns for dealing with associations between classes.

When this PIM based on SOA is created, the PSM of the system can be created based on a target platform such as Web Services, Jini and/or other platforms with transforming tools.

In the next subsections, the illustrative example will be modeled and the PIM based on SOA will be generated using our approach and then this model will be transformed to the PSM based on Web Services (as a chosen platform).

4.1 The PIM of the MCRI System

The PIM of the illustrative example, is showed in figure 3 in this paper, the fragment of the PIM is presented with out other modeling part to simplify the presentation. As shown in figure 3, the MCRI system consists of three classes that implement the main characteristics of the system: *Account*, *Contact* and *Appointment*.

The customer, accesses the *Account* to make a *Contact*, and if it is required then he/she sets an *Appointment*.

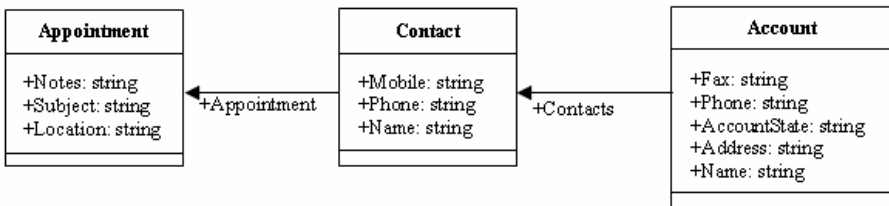


Fig. 3. The PIM of MCRI system

4.2 Generating the PIM Based on SOA

As we mentioned at the beginning of this section, for transforming the PIM to the PSM based on SOA we use a special pattern. As it is shown in figure 4, for the MCRI system three service managers have been designed (AppointmentManager, ContactManager, AccountManager). Each of these service managers has a value object that manages it. For each value object, the following methods are added in the related service manager: Create, Delete, Get and Update. Such as CreateAppointment, DeleteAppointment and ... in AppointmentManager.

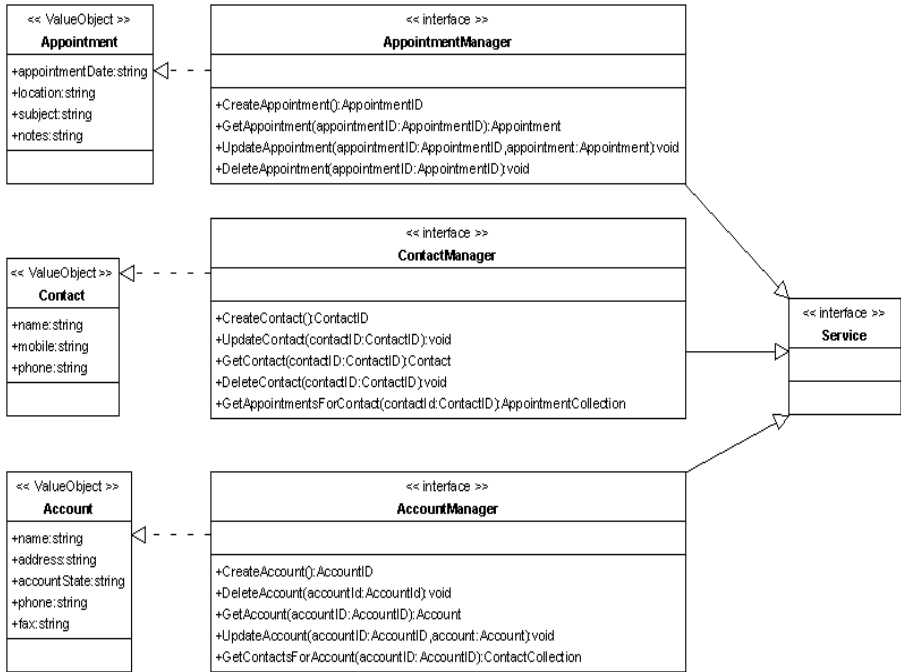


Fig. 4. Generated Service-Oriented Architecture

For each association in the PIM, one or more methods are added to the service manager. For example, GetAppointmentForContact in the ContactManager.

4.3 Generating the PSM Based on Web Services Platform

After generating PSM based on SOA, the final PSM will be designed using a target platform such as Web Services or Jini or In our example, we use Web Services as a target platform.

Web Services are defined by WSDL (Web Services Description Language) [14]. WSDL is an XML-formatted language used to describe a Web service.

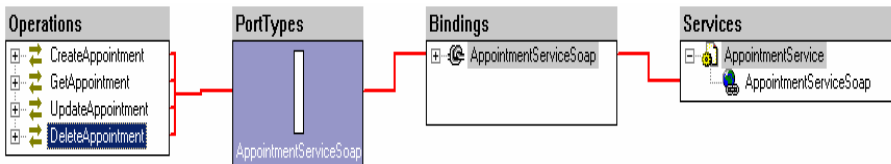


Fig. 5. A part of generated PSM based on WS for AppointmentManager

Transforming PSM based on SOA to the PSM based on Web Services using WSDL is a straightforward task. In our approach, each value object and each Interface in PIM will be transformed to WSDL Type and Port Type in the PSM respectively and the parameters of methods will be transformed to the Messages (Input/Output) in the PSM. For example, a part of generated PSM based on Web Services for AppointmentManager is shown in figure 5.

And its generated WSDL is shown in figure 6. This figure presents the definition of Appointment.

```
<s:complexType name="Appointment">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="AppointmentDate" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Location" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Subject" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Notes" type="s:string" />
  </s:sequence>
</s:complexType>
```

Fig. 6. Definition of Appointment using WSDL

5 Conclusion

In this paper we introduced an approach to modeling and design of complex distributed systems using SOA and MDA. In fact, to exploit the benefits of SOA effectively and duly, we propose an approach that involves MDA into the context.

In this approach the PIM of the system is created and then the PSM based on SOA is generated. Then the final PSM based on a target platform (such as Web Services, Jini and so on) is generated. These models are generated with transformation tools in MDA.

References

1. Jean Bezin, Slimane Hammoudi, Denivaldo Lopes and Frederic Jouault. Applying MDA Approach for Web service Platform. Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conference, EDOC 2004.
2. Michael N.Huhns, Munindar P.Singh, Service-Oriented Computing: Key Concepts and Principles. Journal of IEEE Internet Computing, 2005.
3. Mike P.Papazoglou, Service-Oriented Computing: Concepts, Characteristics and Directions. Proceedings of the Fourth international Conference on Web Information systems Engineering, 2003.
4. S.Cook. Domain-Specification Modeling and Model Driven Architecture. MDA Journal, Pages 1-10, 2004.
5. OMG. Model Driven Architecture (MDA) - document number ormsc/2001-07-01, 2001.
6. Object Management Group. Unified Modeling Language: Superstructure, Apr 2003. Document number: adi2003-04-01.
7. Object Management Group, XML Metadata Interchange (XMI) specification, May 2003. Version 2.0, formal/03-05-02.

8. Markus Debusmann, Reinhold Kroeger, Fachhochschule Wiesbaden, Unifying Service Level Management using an MDA-based Approach , proceedings of the 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Seoul, South Korea, 19.-23. April 2004.
9. Model Driven Architecture – Applying MDA to Enterprise Computing, David S. Frankel – Wiley Publishing, Inc (OMG Press) – 2003
10. Object Management Group. MDA Guide, V1.0.1, omg/o3-06-01, June 2003.
11. Hao He, What is Service-Oriented Architecture?, Orielly WebService Site. <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>, September 2003
12. Qusay H. Mahmoud, Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI), Sun Developers Network, Sun Developers etwork, April 2005
13. Zoran Stojanovic, Ajantha Dahanayake, Henk Sol, Modeling and Design of Service-Oriented Architecture, in proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics, October 10-13, Netherland.
14. E. Christensen, F. Curbera, G. Meredith and S.Weerawarana, Web Services Description Language (WSDL) V.1.1, 2001.

Advanced Data Driven Visualisation for Geo-spatial Data

Anthony Jones and Dan Cornford

School of Engineering and Applied Science, Aston University,
Birmingham B4 7ET, UK
d.cornford@aston.ac.uk
<http://www.ncrg.aston.ac.uk/~cornfod>

Abstract. Most current 3D landscape visualisation systems either use bespoke hardware solutions, or offer a limited amount of interaction and detail when used in realtime mode. We are developing a modular, data driven 3D visualisation system that can be readily customised to specific requirements. By utilising the latest software engineering methods and bringing a dynamic data driven approach to geo-spatial data visualisation we will deliver an unparalleled level of customisation in near-photo realistic, realtime 3D landscape visualisation. In this paper we show the system framework and describe how this employs data driven techniques. In particular we discuss how data driven approaches are applied to the spatiotemporal management aspect of the application framework, and describe the advantages these convey.

1 Introduction

In this paper we describe the basis for an extensible geo-spatial landscape visualisation system that is capable of supporting near photo-realistic rendering in realtime. In particular the framework is designed to allow the type and behaviour within the application to be controlled directly using data driven approaches as well as exposing a set of interfaces which are at a very abstract level and can be implemented as plug-ins. The research brings together elements from modern software engineering, computer games programming and Geographic Information Systems (GIS).

In Section 2 we briefly review the visualisation applications that are currently available and the state of the art in software engineering, especially data driven programming. Section 3 describes the application framework which is based around a central application hub that hosts the connection between a range of services providing the basic functionality of the core system. We describe the main subsystem components and how data driven methods will be applied as part of a scene management system. Finally in Section 4 we discuss the scene and render systems in more detail. We conclude with a summary of possible extensions to the design.

2 Background

Current geospatial representation and management libraries, as well as libraries supplying additional rendering functionality (for example, OGRE¹ and Open Scene Graph²) and commercial applications responsible for producing computer-based renderings based on geographical information (for example, WorldPerfect³ and LandXplorer⁴) offer either limited functionality or are very expensive and often require specialised hardware to run effectively. In contrast many modern computer games place a heavy emphasis on geospatial representation (such as Microsoft Flight Simulator 2004⁵) and these function with increasingly realistic 3D graphics on relatively modest graphics hardware. However the direct application of a game engine to the 3D visualisation of landscape data has several drawbacks. In particular much of the game logic system and Artificial Intelligence would not be appropriate in the visualisation system, and secondly the priorities of the gaming world are different from the user requirements in landscape visualisation.

Geospatial visualisation applications in the field of GIS commonly rely on a low detail data sample of a world model for any given depicted scene. Visualisation data is either extrapolated from the world model (for example, geometry may be an extrusion of a topography layer or the 3D mesh equivalent of a Digital Elevation Model), or sourced from data that is supplementary to the world model such as high resolution satellite imagery. There is little attempt to add detail or clutter in order to improve the immersive quality of the viewer; the focus of such renderings remains the underlying world model, and its emphasis is the information implied by the given data sample. Whereas modern GIS visualisation systems commonly rely on the application of photograph-based textures in order to increase visual quality, we aim to enhance user immersion through the addition of natural entropy and the augmentation of the low resolution data sets upon which GIS often rely.

Another major element of the framework's design is its use of Data-Driven Programming (DDP), which will increase the application's flexibility and extensibility through measured exposure of system functionality. In traditional Object Oriented Programming, objects are described using class definitions, which typify the state and behaviour of the modelled real-world object. Objects with shared state or behaviour are commonly organised into a hierarchy of class inheritance, where child classes exhibit the state and behaviour of their parent types. In DDP, state and behaviour are described separately from their owning objects as components. Each component represents a closely related collection of state and behaviour which together depict a single facet of overall object functionality. Each object thus becomes an aggregation of parameterised components, which together describe the object as a whole. By using data to describe component parameters

¹ <http://www.ogre3d.org/>

² <http://www.openscenegraph.org/>

³ <http://www.metavr.com/products/worldperfect/worldperfect.html>

⁴ <http://www.landex.de/>

⁵ <http://www.microsoft.com/games/flightsimulator/>

and combinations, a class hierarchy can be defined using one or more data files, which collectively drive the logical composition of its constituent run-time objects.

The obvious benefits of DDP are increased extensibility and flexibility; the definition of new object types and behaviours when linked to a scripting language, as well as the modification and instantiation of existing ones, can all be achieved via data manipulation *without access to application code*. In the context of visualising GIS information, the use of DPP enables the user to assemble information rich virtual environments through the combination and extension of existing scene object descriptions. For example, an illustration for visual impact assessment can be quickly tailored to highlight proposed items, add vegetation, buildings and people, or simply increase foreground detail to increase user immersion.

3 The Application Framework

Figure 1 provides an overview of the application framework. The framework's overall design is highly modular, which in turn leads to increased flexibility and extensibility. Loosely based on the design of an object composition framework presented in [1], the framework separates overall application functionality into a number of coherent, loosely coupled responsibilities, each of which is represented in the framework by an abstract interface (illustrated via the inner octagon in Figure 1). The concrete implementation, and thus the run-time behaviour, of each subsystem may be provided by the user in the form of a dynamically linked library⁶ (the outer octagon in Figure 1 denotes such concrete implementations). The behaviour of each subsystem is data driven, with further extensibility provided through the use of application plug-ins⁷. During application execution, the central framework hub performs dynamic allocation and binding of sub-system implementations to their respective interfaces; the hub also acts as an inter-mediating interface between the various subsystems. A brief introduction to the subsystem responsibilities follows.

Assets and Resources

Fundamental framework subsystems represent an asset⁸ processing pipeline, providing an abstraction of the low-level data access and decompression capabilities, upon which a resource⁹ caching sub-system resides. Data access is via a proxy resource handle, which supports one of an enumerated range of basic resource types, including a binary stream, attribute tree, multi-dimensional array, and 3D mesh. A pre-processing stage performs the necessary collection and conversions from original source data to the appropriate asset data format.

⁶ That is, a .DLL file on windows systems.

⁷ A *plug-in* is a portion of code that is compiled into a dynamically linked library – plug-ins commonly extend application functionality via a dedicated exposed interface.

⁸ In the context of the application framework, an *asset* is the optimised data file.

⁹ In the context of the application framework, a *resource* is a run-time equivalent of an asset.

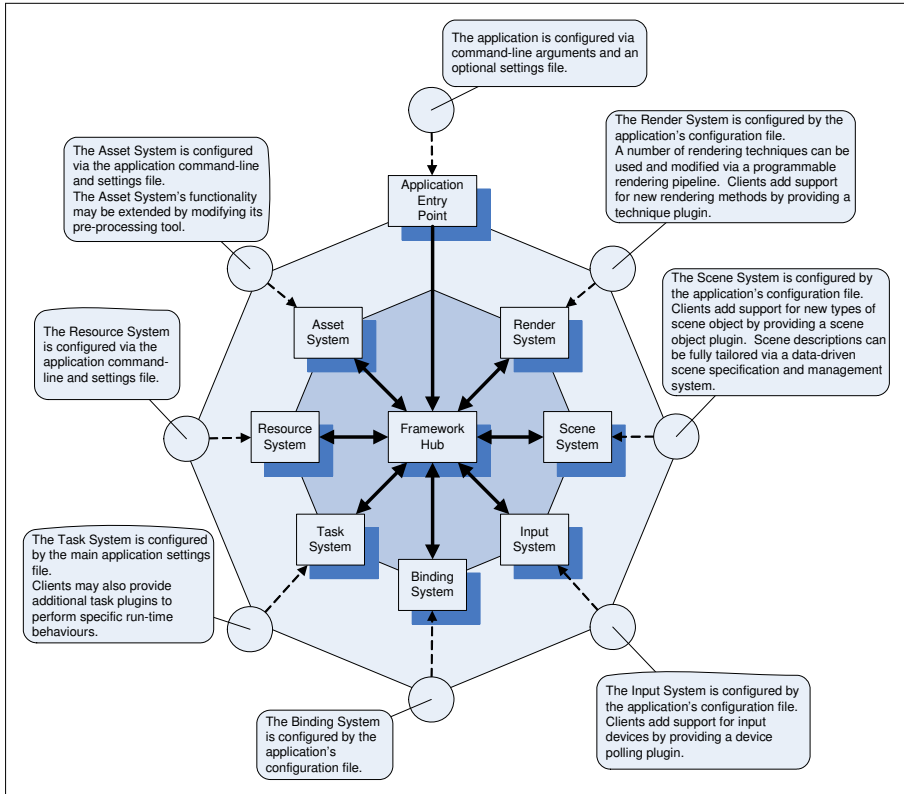


Fig. 1. An overview of the application framework showing the core systems

Run-time Binding

In order to facilitate a data-driven methodology, the application framework includes a flexible and powerful binding system that manages associations between identifiers and memory locations via an information rich run-time type information (RTTI) system. The binding system is responsible for mapping variable identifiers and component attributes to their respective memory ranges; it is also required to manage a registry of exposed variables and functions, and to provide for type-safe, run-time bindings to them. The framework's input system, which is responsible for controlling and monitoring user input via a variety of devices, is a logical extension of the binding system. The input system uses configuration data to bind a given device plug-in, which performs device polling, to one or more run-time variables, whose values are then made available to the remainder of the system.

Application Tasks

The framework's task system manages the application's overall behaviour, which takes the form of an iteration over a number of distinct time slices¹⁰, each

¹⁰ The application framework calls each time slice an *application tick*.

consisting of a multitude of interspersed subsystem operations or events occurring in a given order. Tasks submitted to the task system are ordered and subsequently triggered according to their priority value, although triggering may be deferred until a later time slice, delayed by a given duration, or processed after an absolute time. When triggered, a task is provided with a summary of the task system's status, along with access to the framework hub and hence the application framework as a whole. The task objects themselves are either function objects or function pointers, and are both defined and supplied by subsystem implementations.

Scene Representation and Rendering

The most complex of the framework's subsystems are the scene and visualisation systems, which respectively represent and render a given virtual scene at run-time. The scene system design incorporates a component-based scene graph and dynamically adjusting spatial partitioning system (see Section 4), and will thus embody behavioural, topological and geographical properties of virtual run-time objects. The scene system will also support continuous virtual worlds by streaming scene content in and out of the simulation as required. The render system is designed to make use of the most recent release¹¹ of the OpenGL graphics programming API, including support for both vertex and fragment shader programs. This aims to increase rendering efficiency by minimising render state changes, while maximising visual detail through the use of image-based techniques and effective lighting, shading and shadows.

4 Geospatial Scene Management and Visualisation

Geospatial Scene Management

The problem of runtime representation of a given geospatial virtual scene is comprised of two sub-problems, which coincide with the topological and geometric properties of the scene. While open-source libraries are available to solve these problems (for example OGRE and Open Scene Graph), they tend to be feature rich at the expense of performance, and do not take advantage of emerging methodologies that can greatly increase an application's flexibility and extensibility. The application framework design integrates such strengths by incorporating a novel, performance oriented scene management system that extends contemporary techniques.

In every geospatial scene there exists a hierarchical topological relationship between the scene, its constituent objects, and their properties. The hierarchical relationship can be described using a tree structure that is commonly referred to as a scene graph¹², with nodes representing objects and properties, and arcs denoting the relationships between them (most notably ownership). As scene descriptions become large, the run-time efficiency of storing and manipulating

¹¹ Version 2.0 at the time of writing; a specification is available here:

<http://www.opengl.org/documentation/specs/version2.0/glspec20.pdf>

¹² http://en.wikipedia.org/wiki/Scene_graph

them is reduced; data-driven, component-based methods [2, 3, 4] can be used to counter this. In a scene graph, objects (nodes) have properties with values, which are propagated down to the leaves unless they are overloaded en route by an equivalent property with a differing value. To reduce data duplication and its associated storage and processing overheads, scene objects can inherit from a template or archetype design, with property commonalities stored at the archetype level, and only distinctive values stored at the instance leaves. A component based design lends itself well to DDP¹³, increases flexibility and extensibility as scene object inheritance hierarchies are not necessarily fixed by the software code, and is well suited to the application of design patterns such as the Abstract Factory, Composite, and Prototype patterns [5].

The problem of geometric management is commonly solved with the use of a spatial representation system, which usually takes the form of a hierarchical simplification of the scene through the use of spatial divisions in order to reduce the time complexity of spatial processing such as collision detection and frustum culling. Spatial division algorithms are well-known in the field of GIS (where they often serve an additional purpose in the linearization of a 2D or 3D representation's memory layout); examples include B-trees¹⁴ and Octrees¹⁵, although more advanced algorithms allow for dynamic adjusting of the dividing predicates in order to accommodate scenes containing moving objects [6].

Visualisation

While past incarnations of graphics hardware and software interfaces have utilised a fixed functionality pipeline for transform and rasterisation, today's hardware and APIs have adopted a flexible programmable pipeline that exposes aspects of the geometry and image-based processing functionality to the client. Programs written in a dedicated language, known as shaders, stipulate the appearance of objects in a given virtual scene by specifying tailored transform functions alongside light, material and surface characteristics.

The framework's render system will exploit such exposures by making use of shaders provided as part of data-driven scene descriptions. The resulting combination of shader technology and DDP will allow the user to customise the visual output of simulations to meet their own requirements. We anticipate a particular interest in techniques related to lighting, shading and shadows [7], [8], which have been shown to influence the sense of presence when viewing virtual environments [9].

5 Summary and Future Work

In this paper we have described the basis of a data-driven geo-spatial landscape visualisation system that will be used to support realtime, near photo-realistic

¹³ Component-centric methods actually enforce DDP as properties consist of nothing but data associations.

¹⁴ <http://en.wikipedia.org/wiki/B-tree>

¹⁵ <http://en.wikipedia.org/wiki/Octree>

rendering. We have described the core design of the system, showing how data-driven programming can be used to increase its flexibility, particularly the scene system. We also introduce data-driven programming as a tool for data integration, which remains an open problem for a number of GIS applications [10].

The research is still in progress, and we intend to explore the relation between dynamic data driven approaches to GIS and visualisation solutions and the developing Geography Markup Language (GML), a Resource Description Framework like set of eXtensible Markup Language schema for describing features that exist in the real world¹⁶. The use of GML maps strongly to the data driven aspects of this software, and the sematic structure implied in the GML schema and well written application schema that use GML could facilitate the semi-automatic generation of visualisations using web (feature) services.

References

1. Patterson, S.: An object–composition game framework. In Treglia, D., ed.: *Game Programming Gems 3*. Charles River Media (2002) 15–25
2. Bilas, S.: A data-driven game object system. In: *Game Developers Conference Proceedings*. (2002)
3. Duran, A.: Building object systems - features, tradeoffs, and pitfalls. In: *Game Developers Conference Proceedings*. (2003)
4. Rene, B.: Component based object management. In Pallister, K., ed.: *Game Programming Gems 5*. Charles River Media (2005) 25–37
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley (1995)
6. Luque, R.G., Comba, J.L.D., Freitas, C.M.D.S.: Broad-phase collision detection using semi-adjusting bsp-trees. In: *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, New York, NY, USA, ACM Press (2005) 179–186
7. Wang, J., Sun, J.: Real-time bump mapped texture shading based-on hardware acceleration. In: *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, New York, NY, USA, ACM Press (2004) 206–209
8. Stamminger, M., Drettakis, G.: Perspective shadow maps. In: *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, ACM Press (2002) 557–562
9. Mania, K., Robinson, A.: The effect of quality of rendering on user lighting impressions and presence in virtual environments. In: *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, New York, NY, USA, ACM Press (2004) 200–205
10. Appleton, K., Lovett, A., Snnenberg, G., Dockerty, T.: Rural landscape visualisation from gis databases: a comparison of approaches, options and problems. *Computers, Environment and Urban Systems* **26** (2002) 141–162

¹⁶ The GML standard and associated web feature server specifications can be found at <http://www.opengeospatial.org/>.

Design and Analysis of Test Signals for System Identification

LIU Bo, ZHAO Jun, and QIAN Jixin

Institute of System Engineering
National Key Laboratory of Industrial Control Technology
Zhejiang University, Hangzhou, P.R. China 310027
liubo@iipc.zju.edu.cn

Abstract. For multi-channel process, due to disadvantages of the open-loop single variable step method, multi-channel test method is used. That means all of the channels are tested at the same time. In order to eliminate cross-effect of the different test signals, it requires that all the test signals are uncorrelated. Several test signals are introduced and analyzed. Based on two familiar identification methods: correlation analysis method and least-squares method, we put our strength on the way to get uncorrelated test signals. A novel design for the period length of uncorrelated pseudo random binary sequence (PRBS) is proposed. Use this design method, identifiable PRBS signals can be gained and their periods are the shortest. Simulation results show the effectiveness.

1 Introduction

The fundamental problem in all kinds of predictive control is how to choose an appropriate functional model which reflects the characteristic of the control system. Again, The primary method to acquire a valuable functional model in the current industry is system identification. And the key to obtain a worthwhile result of correct identification is choosing rightly recognizable test signals.

Although multivariable control concept is well accept by industry, the guideline for identification practice is still single variable thinking. Current practice of industry is to use a series of single variable step tests^[1] for model identification, which reflects the fact that each manipulating variable (MV) is stepped separately and some clear step responses are expected for modeling each transfer function. The biggest problem of this test method is its high cost in time and manpower. The product qualities may be disturbed by stepping the MVs. This problem can be solved by using automatic multivariable test approach. But when you input several test signals at the same time, there are some cross-effect between the test signals. For the sake of identifying different impacts on the same output from different inputs in such experiment, these inputs need to be presumed as uncorrelated. Therefore, for multivariable process, we have new design for uncorrelated test signals. Simulation data show that use these design methods, less test time needs and better results can be get.

2 Pseudo Random Binary Sequence (PRBS)

PRBS is a long period signal. In a period, it is a true stochastic binary signal. Over a period, it is a deterministic signal and it will repeat the previous period signal. In the field of system identification, the stochastic of PRBS plays a quite important role.

2.1 Properties of PRBS

A PRBS is a two states signal which can be generated by using a feedback shift register. When the number of registers are n , the maximum possible length is : $N = 2^n - 1$. Assume that the clock period is Δt , the period of PRBS is: $T = N \cdot \Delta t$.

Because the length of PRBS is odd number, in a period, it takes two values $-a$ and a , the number of value “ $+a$ ” is one less than the number of value “ $-a$ ”.

The most characteristic of PRBS signal is its auto-correlation function can resemble the auto-correlation of white noise sequence. For this characteristic, it can be used as test signal. Its auto-correlation function is:

$$R_{xx}(\tau) = \begin{cases} a^2(1 - \frac{N+1}{N} \cdot \frac{|\tau|}{\Delta t}) & -\Delta t < \tau < \Delta t \\ -\frac{a^2}{N} & \Delta t \leq \tau \leq (N-1)\Delta t \end{cases} \tag{1}$$

That is, when N is large enough, in the range $[0, N\Delta t]$, $R_{xx}(\tau)$ can be taken as δ function. See Fig 1.

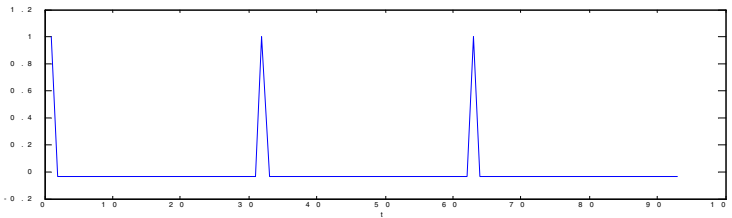


Fig. 1. Auto-correlation function of PRBS

Note that generating a part of a period of a PRBS will not give a signal with the above properties. This makes the selection of time less flexible.

2.2 PRBS Used in Correlation Analysis Method

Firstly for SISO linear system, suppose the input of linear system is $x(t)$, output is $y(t)$. Correlation analysis method uses correlation method to solve the dynamic characteristic of system, that is impulse response $g(\tau)$. The cross-correlation function between input and output is

$$R_{xy}(\tau) = \int_0^{\tau} g(s)R_{xx}(\tau - s)ds \tag{2}$$

Where, T_s is system adjust time, when $\tau \geq T_s$, we get $g(\tau) = 0$. Adjust the period of PRBS, it can be arrived that $T > T_s$.

According to (3), we have

$$g(\tau) = \begin{cases} [R_{xy}(\tau) - R_{yx}(\tau)]/S & \tau \geq \Delta t \\ [R_{xy}(\tau) - R_{yx}(\tau)]/[(\frac{1}{2} + \frac{\tau}{\Delta t}(1 - \frac{\tau}{2\Delta t}))S] & \Delta t > \tau > 0 \\ 2[R_{xy}(0) - R_{yx}(\Delta t)]/S & \tau = 0 \end{cases} \tag{3}$$

where

$$S = (1 + \frac{1}{N})a^2\Delta t \tag{4}$$

Consequently, we can achieve $g(\tau)$ from cross-correlation function between input and output conveniently taking PRBS as input signals. In relative analysis method, used as test signal and in virtue of its periodicity, PRBS does not need infinite time to calculate cross-correlation function in contrast with white noise sequence. PRBS just needs a period time. Therefore to ensure yielding fine identification precision, PRBS shall be easily made and save more time.

As to MIMO system, an old approach is to test every variable independently. Assuming other inputs are zero, we test one selected variable at each time. This approach is time-consuming for one by one test method. Again, it doesn't guarantee to keep other variables intact when checking the selected one. To solve this problem, it needs to make a one and only experiment. But when you input several test signals at the same time, there are some cross-effect between the test signals, that it's difficult to distinguish different impacts on the same output from different inputs. For the sake of identifying different impacts on the same output from different inputs in such experiment, these inputs need to be presumed as uncorrelated.

In order to make uncorrelated PRBS signals, firstly you need to generate a PRBS, secondly shift it horizontally. That is to say, other signals will be this signal's delay-signals. But it still exists a big problem that what is the delay value. To guarantee process dynamic is persistent excited by input signals, the PRBS signal must ensure that its period is longer than the settling time of process for each variable. For this reason, the uncorrelated range of each variable should be longer than its settling time. Therefore, in order to ensure the PRBS signals are uncorrelated and identifiable, what is the minimum delay value we should choose? Aim at this problem we have further study about delay value D .

Theorem 1. Assume a linear process with m-input and n-output, where $n \leq m$, $x_i(t)$ ($i=1, \dots, m$) are input PRBS signals, $x_{i+1}(t)$ are delay signals of $x_i(t)$, suppose the delay value are $K_{i+1}\Delta t$, that is $x_{i+1}(t) = x_i(t + K_{i+1}\Delta t)$. All the input signals are uncorrelated in the range of $0 \sim (N - \sum_{i=1}^m K_i - 1)\Delta t$, where $K_1 = 0$ is the delay value of $x_1(t)$.

Proof. Consider $x_1(t)$ is a PRBS signal, $x_2(t)$ is the delay signal of $x_1(t)$, the delay value is $K_2\Delta t$, so that $x_2(t) = x_1(t + K_2\Delta t)$.

The cross-correlation function between $x_1(t)$ and $x_2(t)$ is

$$\begin{aligned}
 R_{x_1x_2}(\tau) &= \frac{1}{T} \int_0^T x_1(t)x_2(t+\tau)dt = \frac{1}{T} \int_0^T x_1(t)x_1(t+\tau+K_2\Delta t)dt \\
 &= R_{x_1x_2}(\tau+K_2\Delta t) \\
 &= \frac{-a^2}{N} \qquad 0 < \tau < (N - K_2 - 1)\Delta t
 \end{aligned}
 \tag{5}$$

According to (5), the cross-correlation function between $x_i(t)$ and $x_{i-p}(t)$ is

$$\begin{aligned}
 R_{x_i x_{i-p}}(\tau) &= \frac{1}{T} \int_0^T x_i(t)x_{i-p}(t+\tau)dt \\
 &= \frac{1}{T} \int_0^T x_{i-p}(t)x_i(t+\tau+(K_i+K_{i-1}+\dots+K_{i-p+1})\Delta t)dt \tag{6} \\
 &= R_{x_{i-p}x_{i-p}}(\tau+(K_i+K_{i-1}+\dots+K_{i-p+1})\Delta t) \\
 &= \frac{-a^2}{N} \qquad 0 < \tau \leq (N - (K_i+K_{i-1}+\dots+K_{i-p+1}) - 1)\Delta t
 \end{aligned}$$

when N is large enough, $R_{x_i x_{i-p}}(\tau) \approx 0$, so $x_i(t)$ and $x_{i-p}(t)$ will be uncorrelated in the range of $0 \sim (N - (K_i+K_{i-1}+\dots+K_{i-p+1}) - 1)\Delta t$. Thus it can be seen, the uncorrelated range of all signals is $0 \sim (N - \sum_{i=1}^m K_i - 1)\Delta t$.

The theorem shows the uncorrelated range. Through this theorem, it can be known that which is the minimum delay value and we got a new way to design test signals. Consider a linear process with m inputs and n outputs, where $n \leq m$, assume that D_i stands for the settling time for i th input. So that the new design approach of test signals is:

- Step1: Generate a PRBS $x_1(t)$;
- Step2: Calculate $\max D_i (i = 1, \dots, m)$, choose the channel which has the maximum settling time as the first channel and take $x_1(t)$ as its input signal;
- Step3: Choose the remaining input signals for $m-1$ channels stochastically [$x_2(t) \ x_3(t) \ \dots \ x_{m-1}(t) \ x_m(t)$];
- Step4: $x_i(t)$ is the delay signal of $x_{i-1}(t)$, its delay value is $K_i\Delta t = D_i (i = 1, \dots, m)$, where $K_1 = 0$.

From the forward approach, the minimum period of input signals is $T = \sum_{i=1}^m D_i$. In reference [4], the author used maximum settling time of the process as delay value. So its period is $m \times D_{\max}$. By comparing T with $m \times D_{\max}$, especially to a quite complicated

system in which every channel differs in the settling time, it bigly shortens the length of test signal after we make the improvement. The data of example 1 shows it clearly.

3 Generalized Binary Noise (GBN)

This signal was proposed by Tullenken. The motivation was to generate a test signal that is suitable for control-relevant identification of industrial processes.

3.1 Properties of GBN

A GBN signal $u(t)$ takes two values $-a$ and a . At each candidate switching time t , it switches according to the following rule

$$\begin{aligned} P[u(t) = -u(t-1)] &= p_{sw} \\ P[u(t) = u(t-1)] &= 1 - p_{sw} \end{aligned} \tag{7}$$

where p_{sw} is switching probability. The distribution for the event at each switching time is an independent alternative distribution with parameter p_{sw} .

The GBN auto-correlation function is

$$\begin{aligned} R_{uu}(\tau) &= E u(t)u(t-\tau) \\ &= \frac{1}{N} \sum_{t=1}^N a^2 [P(u(t) = a \text{ and } u(t-\tau) = a) + P(u(t) = -a \text{ and } u(t-\tau) = -a) \\ &\quad - P(u(t) = a \text{ and } u(t-\tau) = -a) + P(u(t) = -a \text{ and } u(t-\tau) = a)] \\ &= \frac{1}{N} \sum_{t=1}^N a^2 (1-2p_{sw}) [P(u(t-1) = a \text{ and } u(t-\tau) = a) + P(u(t-1) = -a \text{ and } u(t-\tau) = -a) \\ &\quad - P(u(t-1) = a \text{ and } u(t-\tau) = -a) + P(u(t-1) = -a \text{ and } u(t-\tau) = a)] \\ &= ((1-2p_{sw}) E(u(t-1)u(t-\tau))) \\ &\vdots \\ &= ((1-2p_{sw})^\tau E u^2(t-\tau)) \\ &= (1-2p_{sw})^\tau \frac{1}{N} \sum_{t=1}^N a^2 (P(u(t-\tau) = a) + P(u(t-\tau) = -a)) \\ &= a^2 (1-2p_{sw})^\tau \end{aligned} \tag{8}$$

Especially when: $p_{sw} = 1/2$, then $1 - 2p_{sw} = 0$. Its auto-correlation function is

$$R_{uu}(\tau) = \begin{cases} a^2 & \tau = 0 \\ 0 & \tau \neq 0 \end{cases}$$

From this function it can be seen that GBN auto-correlation function can resemble the auto-correlation function of white noise sequence. And especially when $p_{sw} = 1/2$, the auto-correlation is the same with the auto-correlation of white noise sequence. It can be called white noise GBN. Shows in Fig 2.

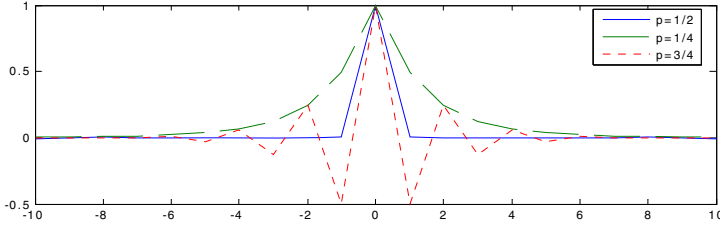


Fig. 2. Auto-correlation function of GBN for different p_{sw}

White noise GBN can be used in relative analysis method. Because its auto-correlation function is δ function, the cross-correlation between any of two white noise GBN signals is zero, that is to say any of two white noise GBN signals are uncorrelated. That is, for MIMO system, we can generate m white noise GBN signals as input signals.

Note that GBN noise has no period, it needs infinite time to calculate cross-correlation function, which needs quite a long time for GBN to attain good precision. Compare with PRBS, PRBS is period signal, it just needs a period time to calculate its auto-correlation function. Therefore, PRBS shall be a better choice to be applied in relative analysis method.

3.2 GBN Used in Least-Squares Method

In least-squares method, to guarantee that the estimation algorithms have unique solutions, test signals should be persistent excitation. PRBS, L signal and GBN can be used in least-squares method.

In a process control environment, a PRBS signal will over-emphasize the high frequency band at the cost of the low and middle frequency band; its frequent fluctuations can be harmful for control actuators. And its power spectrum is

$$\Phi(\omega) = \frac{2\pi\alpha^2}{N} \sum_{k=1}^N \delta(\omega - 2\pi k / N) \quad 0 \leq \omega \leq 2\pi \tag{9}$$

Its spectrum has dips around frequencies $2\pi / \Delta t, 4\pi / \Delta t, 6\pi / \Delta t, \dots$, which will result in low signal-to-noise ratios in these frequency ranges. A PRBS signal is not the best test signal for least-squares method.

A good test signal for process control has a low-pass character. We can generate low-pass GBN signals by reducing the switching probability p_{sw} , or equivalently, increasing the mean switching time ET_{sw} . Define minimum switching time T_{min} as the time (in samples) to keep the signal constant and switching time T_{sw} as the elapsed time between two switches, we can get GBN power spectrum:

$$\Phi_u(\omega) = \frac{(1 - q^2)T_{min}}{1 - 2q \cos T_{min} \omega + q^2} \tag{10}$$

Note that unlike the PRBS signal, the spectrum of a low-pass GBN signal does not have dips at frequencies $2\pi / \Delta t, 4\pi / \Delta t, \dots$, which is advantageous, see Fig 3. So GBN signal is better than PRBS signal in least-squares method.

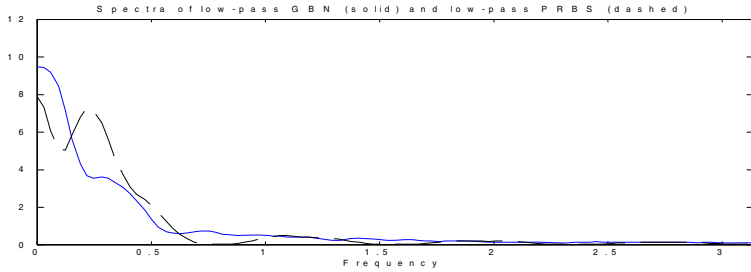


Fig. 3. Compare the power spectra of low-pass GBN and low-pass PRBS

4 Examples

Example 1. Consider a process with three inputs. The transfer functions are:

$$G(s) = \left[\frac{1}{100s+1} \quad \frac{1}{20s+1} \quad \frac{1}{s+1} \right]^T \tag{11}$$

Use correlation analysis method to identify this process. PRBS signals are taken as test signals. Follow the new design steps of PRBS, we get the period of PRBS. Table 1 shows the different period time by using delay value proposed in reference [4] and this paper.

Table 1. Compare the period of PRBS

Process	Settling time	The period of PRBS using the method in reference [4]	The period of PRBS using the method of this paper
Variable 1	500	1500	605
Variable 2	100	1500	605
Variable 3	5	1500	605
The whole process		1500	605

Example 2. Let us see a typical petrochemical processes in refinery industry. There are four main variables. Asymptotic method^[7] extended by least-square method is used here. Choose white noise GBN signals as test signals. Because the uncorrelated character of white noise GBN, we can generate four white noise GBN signals directly and input to the process at the same time. Table 2 shows the duration of one test by using single variable step test and white noise GBN signals.

In multivariable test, as the four variables are test at the same time, the duration time of every variable are the same. Compared to the conventional single variable step test, our design can save about 70% test time. In such a petrochemical process, we can save most cost of time and money.

Table 2. Compare the duration of one test

Process	Settling time	The duration of single variable step test	The duration using multivariable test
Main fractionator	50	50	50
Atmospheric tower	40	40	50
FCC react	20	20	50
Stabilizer	35	35	50
The whole process		>145	50

5 Conclusions

The result of system identification is affected by test signals directly. This article study several kinds of test signals from the point of view of system identification. Firstly, introduce every kind of test signal's attribute. Then analyse how to design uncorrelated test signals in correlation analysis method and least-squares method. In conclusion, due to the periodicity of PRBS, the time to calculate auto-correlation function is comparatively short. Therefore, PRBS shall be a better choice to be applied in relative analysis method than GBN. And we can obtain low-pass signal by converting the convert probability of GBN, such characteristics make GBN works better than PRBS in least-squares method. The future researches will focus more on these test signals more suitable for system identification.

References

- [1] Chongzhi Fang, Deyun Xiao. Process Identification. Qinghua Publishing House. 1988.
- [2] Ljung, L. System Identification: Theory for the User. NJ, Prentice-Hall, UpperSaddle River, 2nd edition, 1999
- [3] M. leskens, L.B.M. Van kessel. MIMO closed-loop identification of an MSW incinerator. Control Engineering Practice , 2002, 10:315–326.
- [4] M.E.H. Amrani, R.M. Dowdeswell, P.A. Payne, etal. Pseudo-random binary sequence interrogation technique for gas sensors[J] . Sensors and Actuators, 1998, B 47:118-124
- [5] Ljung, L. Asymptotic variance expressions for identified black-box transfer function models. IEEE Transactions on Automatic Control, 1985b, 30:834-844.
- [6] Zhu, Y.C. Black-box identification of MIMO transfer functions: asymptotic properties of prediction error models. Int.J.Adaptive Control and Signal Processing, 1989, 3:357-373.
- [7] Zhu, Y.C. Multivariable process identification for MPC: the asymptotic method and its applications. Journal of Process Control, 1998, 8(2):101-115.
- [8] Bainan Li. PRBS signal and relative identification. Science Publishing House. 1987.

The Research on the Method of Process-Based Knowledge Catalog and Storage and Its Application in Steel Product R&D

Xiaodong Gao^{1,2} and Zhiping Fan¹

¹ School of Business Administration, Northeastern University,
Postal Code 11 00 06 Shenyang, China

² Shanghai Baosight Software Co., Ltd, Postal Code 20 12 03
Shanghai, China

gaoxiaodong@baosight.com, zpfan@mail.neu.edu.cn

Abstract. Efficient knowledge storage is for easy to look up and speed up the reaction. Knowledge resource library stored large amount of knowledge achieve, and it can be distill through all kinds of links. Technically, there are two concerns, one is to index through what kind of related strategy, establishes keyword controls, in order to apply the standard literature management tools, the other is how to adopt efficient storage strategies, in order to index and update the knowledge system fast, this thesis takes the operation process as a starting point, investigates the organization strategies towards knowledge classification process and storage, as well as provide a practice case based on research and develop of the steel product.

1 Introduction

Most of the organization works are enclosed by the operation process. Operation process refers to a set of activities which rated together to product values for the customers. Porter describes the operation process of an enterprise as a value chain, competition happens not among the enterprise, but among the value chain of them. Only through efficient management of all the aspects of the value chain, could the enterprise gain the true competition advantage on the market. Processes in organization establish the interactive relationship among the employees, responsibilities, performance and knowledge, through process the enterprise can establish knowledge research system of an organization, and apply it on manufacturing practice and knowledge innovation. The key for its technique is the classification and storage strategy of the knowledge.

2 Knowledge Correlation and Classification

Knowledge correlation refers to the relationship and influence between events, which means to establish the specific relationship based on the different knowledge

requirement. Not a kind of the knowledge innovation deviate from the application of the existing knowledge, In other words is to say the knowledge has the necessity relation with the others, likewise, any kind of the research work needs the other science knowledge as an assistant, without the matting of the other knowledge, the research can not develop. To resolve the series of problems inside an organization requires the related knowledge, background knowledge etc. This related knowledge is what we call the knowledge correlation.

Generally, based on the operation process of the business management program, and strings together the straight forward related knowledge to form a methodical knowledge network, and describe it by language, characters and graphics. Therefore the method of knowledge correlation is the key of the knowledge storage. Figure 1 is the example of the knowledge system of computer maintenance.

Knowledge correlation and knowledge classification have a straight forward relationship. Knowledge classification can be concerned as an infinite and constant development. It is impossible for a rich and colorful world to contain only one classification. Therefore, we can not find only one classification to satisfy the entire knowledge requirement. Knowledge system based on process is also a kind of knowledge classification. Its goal is to operate process and deliver knowledge easily, therefore convenient for knowledge application and innovation.

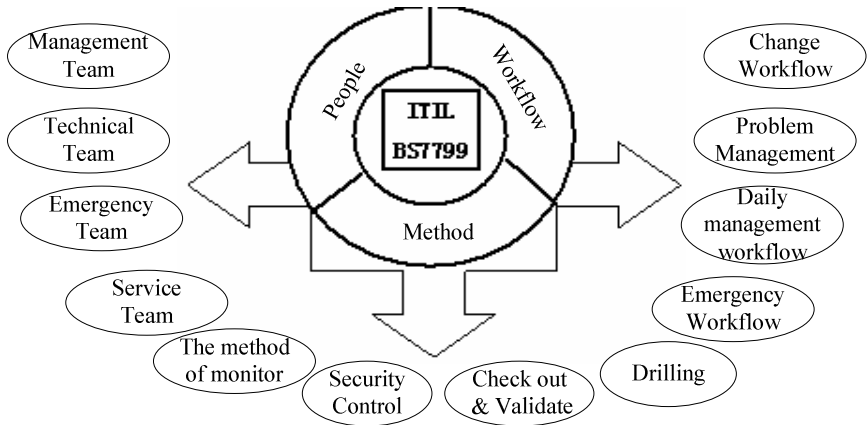


Fig. 1. The knowledge system of computer maintenance

3 Keywords System Based on Process

As all we know, Enterprise’s operation system is based on process. The organic amalgamation among the operation processes consist the operation system of an enterprise. Moreover, Process is consisted by a few working procedures, these procedures are corresponding with working position. In order for the process to be running smoothly, all the working positions need to be cooperate efficiently. Therefore

the operation situation of the whole process is due to the efficiency of all the position. Every position includes the following knowledge elements:

- Performance**, which include time, quality, cost, service, and improvement
- Methodologies**, refers to technical files and working procedures which need to accomplish the target performance
- Hardware**, refers to the working situation and environment which need to support to accomplish the performance
- Communication**, refers to the connections between the up strings and down strings of the employment positions.

Useful knowledge exists in the operation process of an organization. It has been long-term used by the employees and produces the specific benefit. The organic amalgamation of the performance, methodologies, hardware, and communication consist of a knowledge node inside the process, the output information of the position is the input of another, and the amalgamation of all the knowledge (position) forms the knowledge system based on process, Figure 2 is the example of process based knowledge factor.

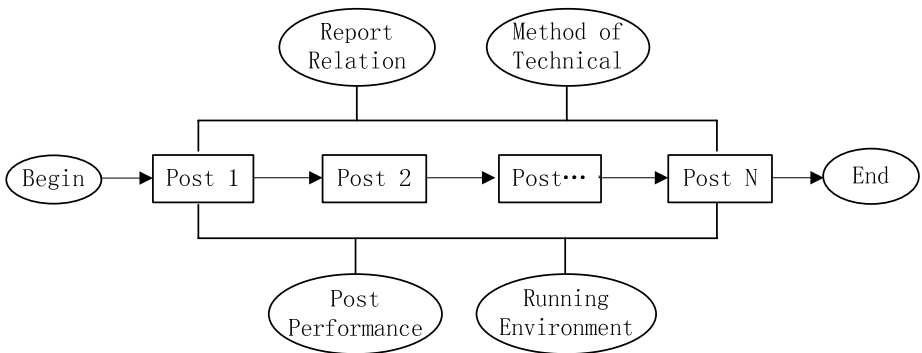


Fig. 2. The sample of process-based knowledge factor

For organization’s efficient knowledge classification, apply a standard archive management tool, a keyword control system needs to be established. This system shall be standard, authorities, and hierarchy. Mostly, 1) keywords dictionary explains and interpret for every keywords, no second definition 2) human resource dictionary based on the keywords of employees identities, includes the personal information , and experience 3) knowledge dictionary based on the relationship between keywords dictionary and knowledge resource 4)keyword correlation dictionary based on the keywords dictionary establishes a multi hierarchy process framework 5) project knowledge correlation dictionary based on keywords, establishes the project knowledge relationship 6) operation knowledge correlation dictionary based on keywords establishes the relationship among the operation knowledge.

4 The Transformation Between Structural Data and Non-structural Data

In information system, technically there are two categories, the structural system and non-structural system. The data processing function in structural system together with the related strategies in non-structural system can be classified in two parts, the information linkage and function linkage 1) information linkage, to have the straight connection for the data inside the database, technically it is very difficult, this needs to transform the data into information, for example to transfer the personal date into the personal information and store them as a non-structural archive. e.g.: Through HTML format transfer every employee's personal data into an information file, and use its employee number as the keywords, this can transfer the structural data into information, and can be applied by the non-structural system.2) function linkage based on the process ability of database, through WEB/EAI technique, this can be the service application.

On the other hand, every knowledge archive needs to obey the archive criterion. For example, Dublin core centralization has its definition for topic and keywords, title, author, abstract and publisher. According to the number of times for the knowledge application, the relatively aged knowledge, its demand for query going down, therefore needs to be rearranged and increase the operation efficiency.

5 The Storage Framework Based on the Knowledge Unit

Knowledge storage strategy can be classified as centralized storage and decentralized storage. It is feasible to utilize the centralized storage for small amount of knowledge storage; it has simple structure, small investment and fast result. But, when the knowledge storage exceeds a certain limit, centralized storage will cause problem, since the large demanding will cause the problem for accessing, therefore speed down the reaction. To spend up the reaction, the system dimension and functional requirements needs to be significantly improved. Therefore increase the investment.

Through practice analysis, it is non-proportional between knowledge innovation and query. Knowledge innovation has much smaller scale than knowledge query. Therefore, to establish a unilateral data synchronized system, through unilateral data source innovation multi-direction data duplication, can guarantee the fast innovation and query of knowledge. A general multi-piece technique product has fast data duplication ability, like IMB, Lotus-Domino system.

Technically, use the micro pc server system to separate knowledge innovation and query, together to support the compression of query, this guarantees the fast reaction system, As fig 3 shows.

Generally, the knowledge construction inside the knowledge center is based on the knowledge field; the enterprise can establish the knowledge warehouse step by step according to its ability, like the manufacturing management knowledge unit in the knowledge operation, and project module knowledge unit in the project knowledge

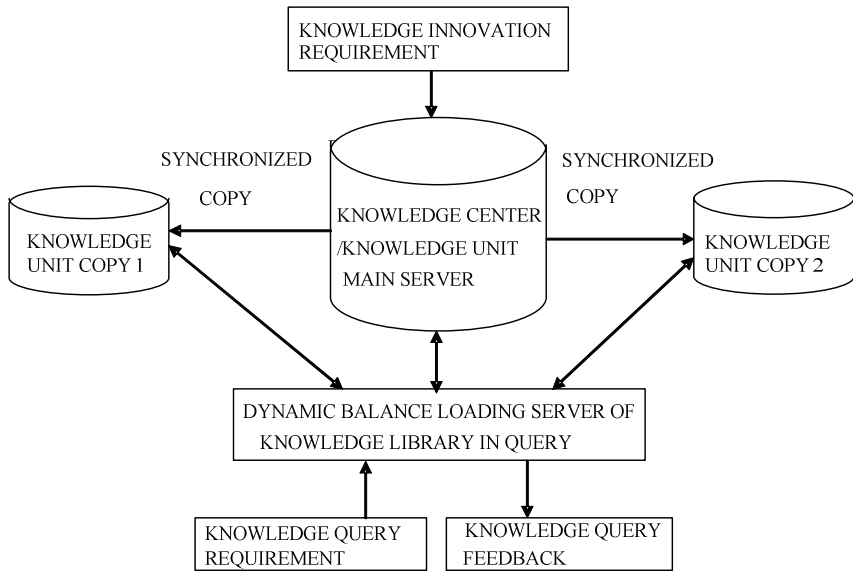


Fig. 3. The overload-balance strategy of knowledge storage

center. Therefore, as the above diagram, we can use the innovation requirement content of one knowledge unit. With the responsible for a server which is the main server, and apply the method of timing innovation, then describe the content to the other by-server,, and by them through the balance loading server, organize the query assignment distribution and feedback to support compressed inquire requirement.

Besides, the by- server of the knowledge unit is the backup system of the knowledge center/ knowledge unit, in case the main server shuts down, the backup system can be switched to be the main server and continue to operate. The main server and the by server, balanced loading server is a logical concept. For a system that has not been frequently visited, it can be configured on a physical device, once the large visits happens, it can be distributed on different physical devices, Thus, apply the micro server, invest a small cost can achieve the flexibility of the system and security of the hardware.

6 Case Application: Process Based Knowledge

Process based knowledge system and its application on the steel product design and development.

The development process of the vehicle plate is based on the specific customer, and organized through a standard recognition process. Therefore, all the project

organization and its related archive classification of the vehicle plate shall be strictly controlled according to three kinds of key techniques (varieties, techniques and application), two key objects (end user, trial-manufacture procedure). Besides, key elements can be increased by the specific project. For example: the type of vehicles, the steel and the number of the projects etc. All these consists the keywords of the vehicle plate knowledge archive.

The so called the high ranking vehicle plate is the kind to satisfy the demand of the inside plate of a medium and high ranking vehicle, it possess the excellent mechanical functionality and cold roll steel as well as the zincification thin steel plate of the surface quality. This is the most elaborate work in the steel product, it has the great difficulty to produce and develop, specifically shows: 1) the large demanding of the vehicle plate. 2) The fast changing of the vehicle style, this cut down the period of changing. 3) The high requirement of the surface plate, this cause the difficulty to manufacture. 4) The automatic pipeline of the vehicle manufacturing which demands the high stability of the steel plate quality. 5) The long time manufacturing workflow and lots of the technical difficulties and high integration.

Through the control system of the key techniques and key objects, together with the visualized techniques to form a visualized vehicle research and development knowledge map. Thereby, the project office can easily choose the related project data for a specific project based on the knowledge map. The main workflow is inside the keywords system of the vehicles plate, according to the specific project keywords. Choose the appropriate project archive through the knowledge system of the technique, product, product application, end user, development workflow, and then deliver the archive to the project and its related product research data, similar technique environment and its research project, the application date of the specific user. Once the new data has been produced, it can be delivered together to the virtual environment of the project.

Abstract the keyword system through professional knowledge, thus the keywords system of a certain profession is consisted by keyword orders (technique order), they interlaced compose a knowledge network. The arrangement of the keyword system in the professional area is the important technique work for the project office. Only to establish the efficient professional keyword system, can one visually understand the development of the professional field and its existing working position (purpose and direction).

7 Conclusion

Knowledge delivery is the new trend of an enterprise's knowledge application and innovation. Through the study of this thesis, we know that knowledge can be organized through process. In order to improve the delivery efficiency, the enterprise needs to establish the dynamic linkage and its protection strategy for of activity, knowledge resource, position/ employee, and apply the knowledge storage system based on the balance loading, resolving scheme.

References

- [1] Portor M. Chen Xiaoyue interpretation, Competition Advantage[M]. Beijing, Publishing Company, 1997.
- [2] Nancy M. Dixon, Common Knowledge --- How Companies Thrive by Sharing What They Know[M]. Harvard College 2000.
- [3] Despres C, Chauvel D. Knowledge Horizons: The Present and Promise of Knowledge Management [M]. Berlin: Butterworth-Heinemann, 2000.

Small WebComputing Applied to Distributed Monte Carlo Calculations

P.A. Whitlock¹, Dino Klein¹, and Marvin Bishop²

¹ Department of Computer and Information Sciences, Brooklyn College
2900 Bedford Avenue, Brooklyn, NY 11210-2889
`whitlock@its.brooklyn.cuny.edu`

² Department of Mathematics/Computer Science, Manhattan College
Riverdale, New York 10471
`marvin.bishop@manhattan.edu`

Abstract. The software package, Small WebComputing (SWC), has been applied to a Monte Carlo simulation of a system of hard hyperspheres in a variety of dimensions. The SWC environment was chosen because once the framework is embedded in the application code, the user has the choice of running the distributed computations as a set of applets, as parallel threads on a symmetric multiprocessor or as independent processes distributed over a network. A description of the software and a discussion of its ongoing evolution is presented.

1 Introduction

The properties of hard hyperspherical systems in a variety of dimensions have been studied by us using Monte Carlo methods[1, 2, 3]. In order to study systems with a minimum of several thousand hyperspheres a parallel or distributed computation was employed. The Small WebComputing (SWC) protocol developed by Ying and co-workers[4, 5, 6] was selected for the project because of several attractive features. Firstly, by simply incorporating several new classes into the Java simulation code, the actual distribution of the code is performed by the SWC software. The user is only responsible for the division of the calculation into tasks that can be run in parallel. Secondly, the distributed tasks can be run as Java applets and the sole requirement is that the host computer has the appropriate Java libraries, i.e. there is no need to upload the complete SWC software onto every host. Thirdly, the distributed tasks can also be run as threads or conventional distributed processes depending on the computers available.

An overview of the existing SWC software, a description of the Monte Carlo application, and a discussion of the ongoing improvements are given in this paper.

2 The Original SWC Software

The Small WebComputing framework was developed to provide master-worker MIMD parallel programming software for a user[5]. It was meant to have an

easy to use design that separated the programming interface from the underlying hardware. To achieve its goals, the SWC software was written in Java. The use of Java and Java applets provided inherent security through the "sandbox" security model[7, 8]. A set of extensible classes that sustain the framework for the user's computation were provided. In the original version of SWC, upper level communication, e.g. between the Master process and the server, used the Transmission Control Protocol (TCP) and lower level communication used the User Datagram Protocol (UDP). Eager scheduling[9, 10] was employed to guarantee load balancing and task completion.

The SWC framework envisions a Computation as composed of three levels: the Master, the Router, and the Worker components. A Computation that utilizes the original SWC framework must contain a Master, one or more Routers, and one or more Workers. The user programs the Master component so that it divides the computation into smaller, independent tasks, referred to as Work Units. The Workers are responsible for processing the Work Units (completing the task that the Work Unit represents) and returning Result Units to the Master. The Router component is provided by the software and its purpose is to act as a liaison between the Master and the Workers. All communication with the Workers is channeled through the Routers which protects the Master from being flooded by messages from Workers. Thus, the communication is tiered, with the Master only communicating with the Routers, the Routers communicating with both the Master and the Workers, and the Workers communicating only with the Routers (no inter-Worker transmission of data is allowed). Under eager scheduling Computations are guaranteed to finish. Any incomplete task, *i.e.* no Result Unit has yet been returned for the task, is kept on the task queues by the Routers and reassigned to idle Workers that request Work Units. The SWC software is completely thread based and therefore can easily take advantage of multiple CPU host computers.

In order to create a computation, the user must extend and implement the following abstract classes and interfaces:

SWCWorkUnit: a class implementing this interface will contain data regarding the task that needs to be done

SWCResultUnit: a class implementing this interface will contain results from the completed task

SWCMaster: the implementation of this abstract class must generate Work Units, collect Result Units, and output final computation results

SWCWorker: the implementation of this abstract class processes the Work Unit and creates a Result Unit which is returned to the Master.

It is important to note that the concrete class, SWCRouter, which functions as the Router in the system, does not need to be extended since it does not perform any computation specific operations. In the original version of SWC, the Router could not be shared among multiple computations because a Router was connected to one Master managing a single Computation.

3 Small WebComputing 2: An Improved Version of the Software

The second version of the Small Web Computing (SWC2) framework[11] improved the usability of the software. Computation specificity was completely removed from the Master and Worker, and the framework hosted multiple, independent, concurrent Computations. The Router's functionality was still confined to transferring Work or Result Units among the Master and the Workers. This lack of specificity was achieved by incorporating a "pull model" of communication between each level of the framework. A Worker process requested (pulled) a Work Unit from a Router, while the Router requested Work Units from the Master. Computation completion was still guaranteed through eager scheduling.

SWC2 employed TCP for all communication between the components. The original SWC framework relied on UDP as the protocol for communicating with Workers. This choice, coupled with the inability to fragment Work or Result Units at the framework level, limited the size of Work or Result Units to 64 kilobytes. The decision to use UDP may have been due to the implementation of the TCP classes in the JDK version current when the SWC framework was originally written. In Table 3 of Ying, Arnow and Clark[4] which gives elapsed times for 100 iterations of transmissions of various data sizes, the TCP tests failed to transmit 4K and 64K data instances. Another difficulty with the use of UDP was the unreliable delivery of Work Units and Result Units. The SWC framework contained code to check periodically for the delivery of UDP packets. This code was the source of timing bottlenecks and bugs which hindered the early usages of the software. Since TCP guarantees delivery, the SWC2 framework does not need to test for delivery and perform retransmission of Work and Result Units. These changes simplified the software and lead to greater transparency of the SWC2 source code. There is no longer a restriction on the size of the data transmitted. Exhaustive testing has shown that there is no problem with using TCP for large data sets or frequently sent transmissions in SWC2.

4 The Monte Carlo Application

The behavior of multi-dimensional hyperspherical systems is governed by the Boltzmann distribution function, $f(\mathbf{R})$:

$$f(\mathbf{R}) = \frac{\exp[-\sum \phi(r_{ij})/k_b T]}{\int \exp[-\sum \phi(r_{ij})/k_b T] d\mathbf{R}} \quad (1)$$

where \mathbf{R} is the d-dimensional vector, $\mathbf{r}_i = (x_{i1}, x_{i2}, \dots, x_{id})$, $i = 1, \dots, M$ of the d*M coordinates of the centers of mass of the M hyperspheres in the simulation box, k_b is Boltzmann's constant and T is the absolute temperature of the system. The pair potential, $\phi(r_{ij})$ represents the interaction between two hard hyperspheres; $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ and

$$\phi_{hs}(r_{ij}) = \begin{cases} \infty, & \text{when } r_{ij} < \sigma \\ 0, & \text{when } r_{ij} \geq \sigma \end{cases} \quad (2)$$

where σ represents the contact distance between the two hyperspheres or the hypersphere diameter. The hyperspheres are initially placed in a simulation box on a lattice. The side lengths of this box are determined by the number of hyperspheres and the number density, ρ , the number of particles per unit volume.

The Metropolis [12] method was developed to assist in the Monte Carlo evaluation of integrals associated with complex physical systems. It guarantees the asymptotically correct sampling of an integrand, interpreted as a probability distribution function, by performing random walks in the many dimensional configuration space of the problem. These walks are generated by proposing a move from the current position of a hypersphere, \mathbf{X} , to a new position, \mathbf{X}' . The new position is chosen from a probability distribution function, $H(\mathbf{X}' | \mathbf{X})$. In the present calculation, the new position is randomly chosen from a hyperbox surrounding the current position of the center of mass of the hypersphere. The proposed new position is then accepted or rejected based upon the probability $p(\mathbf{X}' | \mathbf{X})$ defined as:

$$p(\mathbf{X}' | \mathbf{X}) = \min(1, q(\mathbf{X}' | \mathbf{X})) \tag{3}$$

where

$$q(\mathbf{X}' | \mathbf{X}) = \frac{H(\mathbf{X} | \mathbf{X}')f(\mathbf{X}')}{H(\mathbf{X}' | \mathbf{X})f(\mathbf{X})} \tag{4}$$

If the new position is not accepted, the hypersphere remains at its current location. The acceptance ratio, the number of accepted moves divided by the number of total moves, is monitored. At higher densities, it is necessary to use much smaller trial moves in order to have a “reasonable” configuration change. Each pass of the random walk consists of one attempted move for each of the M hyperspheres. The move may or may not be accepted but is always counted in the averaging. As the random walk proceeds, a recursive relationship develops between the phenomenological distribution functions, $f_n(\mathbf{R})$, represented by each step of the random walk. As long as the system is ergodic and obeys detailed balance [12, 13, 14], $f_n(\mathbf{R}) \rightarrow f(\mathbf{R})$ is guaranteed to be true as n , the number of passes, becomes large.

The Monte Carlo averaging procedure is performed to compute expectation values, $\langle A \rangle$, of physical properties, $A(\mathbf{R})$:

$$\langle A \rangle = \frac{\int A(\mathbf{R})f(\mathbf{R})d\mathbf{R}}{\int f(\mathbf{R})d\mathbf{R}}. \tag{5}$$

In the Metropolis algorithm the successive positions of the hyperspheres are not independent; it takes many passes to converge from the initial state to an equilibrated one sampled from $f(\mathbf{R})$. Thus, some number of passes in the random walk must be discarded. Typically, on the order of $10^3 - 10^4$ passes are needed to reach the equilibrated state. Once the asymptotic distribution function is sampled, there is still serial correlation between each step in the random walk and this will affect the determination of the statistical error of the results. One method of dealing with these statistical correlations is to divide the random walks

into blocks and use the block averages in the error analysis. Another approach is to perform totally independent sets of random walks and average the individual results together. The latter method is ideally suited for implementation with the SWC software.

Ying and co-workers[5] original vision was that the programmer would divide their computation into small tasks which would require an hour or two on any computer. This would not burden the volunteer host as it would likely be available for this amount of time. In the present case, for lower dimensional systems such as one or two dimensions, a random walk converges in one or two hours. The time for calculating a converged random walk increases dramatically in higher dimensions. A four dimensional system with 4096 hyperspheres typically took six hours and forty one minutes to complete ten parallel instances of a random walk consisting of thirteen thousand passes. However, an identically sized system in six dimensions, 4096 six dimensional hyperspheres, took thirty three hours to complete ten parallel instances of a random walk of eleven thousand passes at a similar number density. These large times for individual tasks require processors which are available for long periods of time and such tasks are probably not suited for running on widely scattered volunteer hosts.

While some of the calculations of the hard hypersphere systems have been deployed over a heterogeneous set of computers via the Internet, most of the production runs have been carried out on a network of Sun workstations. This mode of computation was selected because it was easier to harness a large group of networked workstations than to access a large number of volunteer computers. When volunteer hosts are requested to run the applet tasks, the version of the Java Virtual Machine running on each machine can be an issue. However, the applets did run successfully on PCs using Internet Explorer or Netscape running Windows, PCs using Mozilla running Linux and PCs using Netscape running Solaris. In the setting of separated colleagues collaborating on a calculation and thus able to negotiate the choice of browser and JVM, the applet version of the Worker is a feasible method of performing calculations.

When a computationally intensive code is written in Java, there is always a concern that the calculations will run substantially slower than if they were implemented in a different language. Extensive testing was done to compare the Java simulation code using the original SWC libraries against a serial C++ code. The serial C++ code and the parallel SWC code were run on the same processor and gave the exact same results. In a comparison of two, five dimensional runs involving 3125 hard hyperspheres with 3000 total passes, the serial code took 2 hours and 24 minutes, whereas the Java code with just one Worker took 2 hours. Similar timing results were obtained in all test cases, confirming our decision to use the SWC software.

5 Evolution of SWC2 to Version 3

After several years of experience with the software, major improvements are needed. In the SWC2 organization, no communication can occur directly between

Workers. Each task has to be independent of the other tasks. To extend the hypersphere computation to dimensions greater than eight will require a major change in the way the the random walks are performed. The easiest alteration would be to divide each of the parallel random walks into tasks representing a thousand or so passes and to then serialize the random walk into sets of tasks. This would require the transmission of the intermediate positions of the hyperspheres in the random walk, between the Master and the Worker, to continue the walk. This can be accomplished with SWC2. An example of organizing a computation to allow for multiple exchanges of data between the Master and the Workers is given by the sample factorial computation in the SWC2 documentation[11]. For very large numbers of particles, a commonly used alternative is to divide the domain space of the hard hypersphere system into subdomains containing sets of neighboring hyperspheres[15, 16]. In the present Computations, the domain space is already partitioned into hypercells in order to efficiently detect hypersphere overlap. In a domain decomposition paradigm, the subdomains would represent distributed tasks and could be run on different Workers. If a hypersphere moved sufficiently far, it would need to be transferred to a different subdomain. Since the originating Worker would not know the location of the receiving Worker, a broadcast to all Workers might be needed. Alternatively, a barrier might be necessary to ensure that the Workers are approximately all at the same step in the random walk before a hypersphere is transferred. Workers would then need to check for a message at every move[17]. This is a well-known problem in parallel simulations[18] and can lead to great inefficiencies in the simulation[19].

The most fundamental change in the newest version, SWC3, is the communication system. It is being modified from a "pull model" to a fully asynchronous system. Messages can be sent between the Master and the Workshops, which manage workers. These messages can be control messages or other kinds of messages sent among Computations and Workshops.

The system allows, to limited extent, for communication to occur between a Computation and its Workshops, or among the Workers within a given Workshop. Restrictions on the type of communication are required in order to sustain the assumption that Work Units processed by different Workers, at different points in time, will produce results that are statistically identical.

These new communication capabilities are supported by the Router component, which exhibits enhanced functionality and responsibility. The Routers in the SWC3 system now communicate among themselves in order to locate Workers for inter-Worker messaging. Since reliability is not required for inter-router queries, the faster and more compact UDP protocol is used. Worker messages are still reliably sent using TCP connections.

Another benefit of the new communication system using control messages is the ability to terminate Workers on demand. In the previous version, SWC2, it was impossible to terminate specific Workers. If a Computation finished, but some of its replicated Work Units were still being processed, the user would have to wait needlessly. The user's alternative was to terminate the SWCWorker process, with the risk of negatively impacting another Computation. When a

Computation terminates in SWC3, a notification message is sent to all Routers, which is subsequently transmitted to all Workshops. The notification causes Routers to dispose of any Work Units created by the now completed Computation, while Workshops will terminate any associated Workers.

Finally, a new interface is being developed in which Computations may periodically checkpoint, *i.e.* save a snapshot of their state. The purpose of this new functionality is to reduce the amount of lost work in the event that the Master terminates unexpectedly because if a Master is deleted, all the Computations are lost.

6 Conclusion

The Small WebComputing framework has been successfully used in the investigation of large hypersphere systems distributed on local area networks. To achieve the next stage of the investigation, dimensions higher than eight and tens of thousands of hyperspheres, requires changes in the organization of the computation code. How well this will work over the Internet with its inherent communication latency and a higher frequency of lost tasks needs careful evaluation.

Acknowledgments

We wish to thank the Brooklyn College Computing Center and the Manhattan College Computing Center for their support. One of us, D.K., was partially supported by PSC/CUNY Award # 65358-0036.

References

1. Whitlock, P.A., Klein, D., and Bishop, M.: A parallel Monte Carlo Simulation of a 5-Dimensional hard sphere system using SWC. CUNY Ph.D. Program in Computer Science Technical Report, TR-200405, <http://www.cs.gc.cuny.edu/tr/>
2. Bishop, M., Whitlock, P.A. and Klein, D.: The Structure of Hyperspherical Fluids in Various Dimensions. *J. Chem. Phys.* **122** (2005) 074508
3. Bishop, M. and Whitlock, P.A.: The Equation of State of Hard Hyperspheres in Four and Five Dimensions. *J. Chem. Phys.* **123** (2005) 014507
4. Ying, K., Arnow, D. and D. Clark: Evaluating Communication Protocols for Web-Computing. In Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications, CSREA Press, Las Vegas, June (1999)
5. Arnow, D., Weiss, G., Ying, K. and Clark, D.: SWC:A Small Framework for Web-Computing. In Proceedings of the International Conference on Parallel Computing, Delft, Netherlands, August (1999)
6. Ying, K.M.: WebComputing: Design and Performance. Ph.D. dissertation, Computer Science, City University of New York (2000)
7. <http://java.sun.com/sfaq/verifier.html>
8. McGraw, G. and Felten, E.W.: Securing Java: Getting Down to Business with Mobile Code, 2nd Edition. John Wiley & Sons, New York, (1999) Chapter 2.

9. Baratloo, A., Karaul, M., Kedem, Z.M. and Wijckoff, P.: Charlotte: Metacomputing on the Web. *Future Generation Computer Systems*. **15** (1999) 559
10. Neary, M.O. and Cappello, P.: Advanced Eager Scheduling for Java-Based Adaptively Parallel Computing. In *Proc. of the 2002 Joint ACM - ISCOPE Conference on Java Grande (2002)* 56
11. <http://www.sci.brooklyn.cuny.edu/~whitlock/swc2docs/>
12. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E.: Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21** (1953) 1087
13. M.H. Kalos and P.A. Whitlock, *Monte Carlo Methods*, John Wiley and Sons, Inc., New York, 1986, pp. 73–86.
14. W.W. Wood, Monte Carlo studies of simple liquid models, in: H.N.V. Temperley, J.S. Rowlinson and G.S. Rushbrooke, (Eds.), *The Physics of Simple Liquids*, North-Holland, Amsterdam, 1968, Chapter 5.
15. Barnes, J.E. and Hut, P.: A Hierarchical $O(N \log N)$ Force Calculation Algorithm. *Nature* **324** (1986) 446–449
16. Greengard, L. and Rokhlin, V.: A Fast Algorithm for Particle Simulations. *J. Comp. Phys* **73** (1987) 325–348
17. Wilkinson, B. and Allen, M.: *Parallel Programming*, Second Edition. Pearson Prentice Hall, Upper Saddle River, NJ (2005) Chapter 6
18. Jefferson, D.R.: Virtual Time, *ACM Transactions on Programming Languages and Systems* **7** (1985) 404–425
19. Jones, K. and Das, S.R.: Combining Optimism Limiting Schemes in Time Warp Based Parallel Simulations. In: D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds., *Proceedings of the 1998 Winter Simulation Conference*

Monte Carlo Grid Application for Electron Transport*

Emanouil Atanassov, Todor Gurov, Aneta Karaivanova, and Mihail Nedjalkov

IPP - Bulgarian Academy of Sciences,
Acad. G. Bonchev St., Bl.25A, 1113 Sofia, Bulgaria
emanouil@parallel.bas.bg, gurov@parallel.bas.bg, anet@parallel.bas.bg,
mixi@parallel.bas.bg

Abstract. In this paper we present a Grid application developed for electron transport problems called **SALUTE** (**S**tochastic **A**lgorithms for **U**ltra-fast **T**ransport in **s**emiconductors). We consider a physical model of a femtosecond relaxation of optically excited electrons which interact with phonons in an one-band semiconductor. The electron-phonon interaction is switched on after a laser pulse creates an initial electron distribution. The Barker-Ferry equation is utilized as a quantum-kinetic model of the process under consideration. Two cases of this process are investigated - with and without an applied electric field. The electric field causes shift in the replicas, population of the semiclassically forbidden regions and influences the broadening and retardation of the electron distribution. The paper describes Grid implementation of these CPU-intensive algorithms. Using this application innovative results for different materials can be obtained. Here we present the first version of SALUTE which is used to obtain innovative results for GaAs materials. The results from a number of tests on MPI-enabled Grid are shown and discussed.

Keywords: Monte Carlo methods, electron transport, Markov chains, parallel computing, Grid applications.

1 Introduction

Monte Carlo Methods (MCMs) are based on the simulation of stochastic processes whose expected values are equal to computationally interesting quantities. MCMs offer simplicity of construction, and are often designed to mirror some process whose behavior is only understood in a statistical sense. However, there is a wide class of problems where MCMs are the only known computational methods of solution. It is well known that MCMs form the computational foundation for many fields including transport theory, quantum physics, computational chemistry, finance, etc.

* Supported by the Ministry of Education and Science of Bulgaria under Grant No. I1405/04, by Austrian Science Fund under FWF Project START Y247-N13 and by FP6 INCO Grant 016639/2005 Project BIS21++ .

In this paper we describe a Grid application, which solves a computationally intensive problem arising from semiconductor physics (see Section 2). The underlying stochastic algorithms are of Monte Carlo type. The application is called SALUTE for simplicity.

When using a MC method, a set of computational random numbers has to be generated and used to statistically estimate a quantity of interest. The probabilistic convergence rate is known to be approximately $O(N^{-1/2})$, where N is the number of underlying random samples. A serious drawback of the MC methods is the slow rate of convergence. In order to accelerate the convergence rate of the MC methods several techniques have been developed. Variance reduction methods, like antithetic varieties, stratification and importance sampling [1], reduce the variance which is a quantity to measure the probabilistic uncertainty. Another way to improve the convergence is to use quasi-random sequences instead of pseudorandom. While pseudorandom numbers are constructed to imitate the behavior of the truly random numbers, quasi-random sequences are deterministic and constructed to be as uniform as possible at the cost of correlation.

Parallelism is an alternative way to accelerate the convergence of a MC computation. If n processors execute n independent copies of a MC computation, the accumulated result has a variance n time smaller than that of a single copy. Due to the nature of MC methods, it is sometimes appropriate to have as many executions as possible to reduce the statistical error. On the other hand, for a distributed MC application: once a distributed task starts, it can usually be executed independently with almost no inter-process communications. Therefore, MC applications are widely perceived as computationally intensive but naturally parallel. The subsequent growth of computer power, especially that of the parallel computers and distributed systems, made distributed MC applications possible to carry out more and more ambitious calculations [2, 3]. Actually, many MC applications and software packages in science and engineering, for example, CHARMM [4] for macromolecular dynamics simulation implemented using MPI, PMC [5] for nuclear physics simulation on Livermore Message Passing System, and MESYST for simulation of 3D tracer dispersion in atmosphere running on a CRAY T3E parallel machine, have already taken advantage of the power of parallel systems to achieve a more accurate understanding of the problem or better performance of the computation.

Compared to the parallel computing environment, a large-scale distributed computing environment or a Computational Grid [6, 7, 8] has tremendous amount of computational power. Let us mention the EGEE Grid which today consists of over 18900 CPU in 171 Grid sites¹.

¹ The Enabling Grid for E-science (EGEE) project is funded by the European Commission and aims to build on recent advances in grid technology and develop a secure grid infrastructure which is available to scientists 24 hours-a-day. The project aims to provide researchers in both academia and industry with access to major computing resources, independent of their geographic location. the EGEE project identifies a wide-range of scientific disciplines and their applications and supports a number of them for deployment. For more information see <http://public.eu-egee.org/>.

There are already MC applications running on the Grid (see some of the EGEE applications), and also publications like [9, 10, 11], for example, considering MCMs and large distributed computing paradigms.

Effectively exploring the power of distributed MC application requires that the underlying random number streams in each subtask are independent in a statistical sense. The main techniques used in parallel random number generators to distribute sequentially generated random number sequence among different processors include sequence splitting and leapfrog. One problem with the sequence splitting and leapfrog techniques is that we must either assume that the number of nodes is fixed or at least bounded which will restrict the scalability of the distributed MC computations. Another way to generate parallel random number sequences is to produce independent sequences by properly parameterizing pseudorandom generators [12]. The SPRNG (Scalable Parallel Random Number Generators) library [13] was designed to use parameterized pseudorandom generators to provide independent random streams. Some generators in SPRNG can generate up to $2^{31} - 1$ independent random number streams with sufficiently long period and good quality [13]. These generators meet the requirements of the most grid-based MC applications.

The intrinsically parallel aspect of MC applications makes them an ideal fit for the grid-computing paradigm. In general, grid-based MC applications can divide the MC task into a number of subtasks (called jobs in EGEE Grid) by the *task-split service* and utilize the grid's *Workload Management Service* (also called *Resource Broker* in EGEE) to dispatch these independent subtasks to different nodes [14]. The *connectivity services* are provided using the *gridftp* protocol. The execution of a subtask takes advantage of the Storage Elements to store the executable, intermediate results, and to store each subtask's final (partial) result. When the subtasks are done, the EGEE *Workload Management Service* and the *Logging and Bookkeeping Service* are used to collect the information about the results and then the *task-gathering service* gets the results of the successfully executed subtasks. We have prepared a task-split service and a task-gathering service using the SQL language. The paper is organized as follows: After an overview (the Introduction) we describe the application (section 2), then we discuss the parallel and Grid implementation of SALUTE, and, finally, we give some conclusions and directions for future work.

2 Description of the Application

Many problems of interest in transport theory and related areas can be cast as a Fredholm integral equation of the second kind [1]. It is instructive to rewrite such equation as:

$$f = Kf + \varphi, \tag{1}$$

where K is an integral operator, and f , φ are functions in a Banach space of integrable functions.

The equation (1) shows that the kernel, K , appears as a linear operator. MCMs are merely probabilistic ways of applying linear operators. The formal

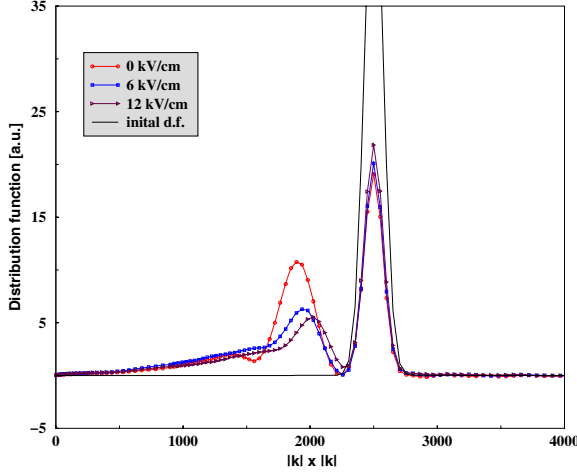


Fig. 1. MC solution at positive direction on the z axis ($t=200$ femtoseconds)

solution of the equation is presented as a Neumann series with terms obtained by consecutive application of K . The general MCM problem is to calculate linear functionals of the solution:

$$J(f) = \int_D h(x)f(x)dx = (h, f), \quad (2)$$

where h a given function. In particular, the MCM approach consists of the following steps (1) to define a suitable random variable whose mathematical expectation is equal to the solution, (2) to determine appropriate initial and transition probability densities by using the kernel of the integral operator after some necessary transformations for the chosen random process, (3) to apply various variance reduction techniques to reduce the statistical error, and, finally, to formulate the appropriate statistical estimator.

MCMs for quantum transport in semiconductors and semiconductor devices have been actively developed during the last decade [15, 16]. If temporal or spatial scales become short, the evolution of the semiconductor carriers cannot be described in terms of the Boltzmann transport [17] and therefore a quantum description is needed. We note the importance of active investigations in this field: nowadays nanotechnology provides devices and structures where the carrier transport occurs at nanometer and femtosecond scales. As a rule quantum problems are very computationally intensive and require parallel and Grid implementations.

A Wigner equation for the nanometer and femtosecond transport regime has been derived from a three equations set model based on the generalized Wigner function [18]. The full version of the equation poses serious numerical challenges. A simplified version of the equation (in the homogeneous case - the Levinson or Barker-Ferry equation, [19]) is analyzed in the framework of SALUTE. The physical model describes a femtosecond relaxation process of optically excited

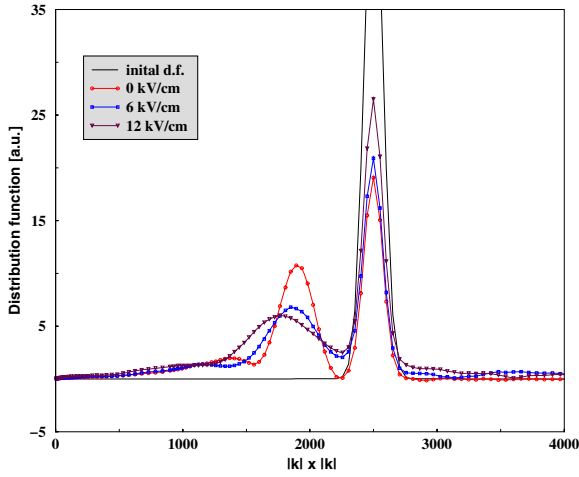


Fig. 2. MC solution at negative direction on the z axis ($t=200$ femtoseconds)

electrons which interact with phonons in single-band semiconductor. The interaction with phonons is switched on after a laser pulse creates an initial electron distribution.

Experimentally, such processes can be investigated by using ultra-fast spectroscopy, where the relaxation of electrons is explored during the first hundred femtoseconds after the optical excitation. In our model we consider a low-density regime, where the interaction with phonons dominates the carrier-carrier interaction. Two cases are studied using SALUTE: electron evolution in presence and in absence of electric field. The electric field causes shift in the replicas population of the semiclassically forbidden regions and influences the broadening and

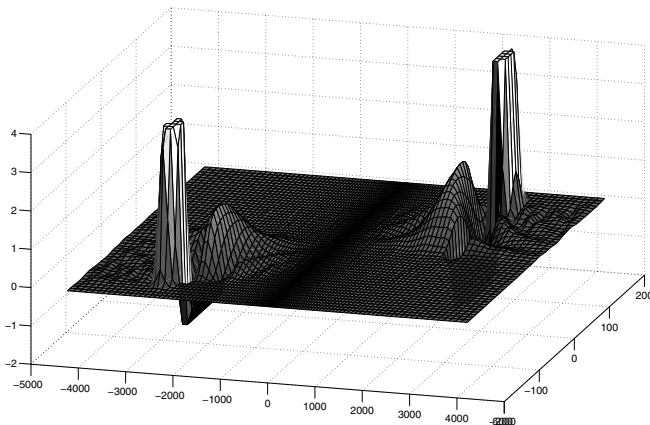


Fig. 3. Distribution of optically generated electrons in a quantum wire

retardation of the electron distribution (see Figures 1-3). The intracollisional field effect is clearly demonstrated as an effective change of the phonon energy, which depends on the field direction and the evolution time.

Another formulation of the Wigner equation considers inhomogeneous case when the electron evolution depends on the energy and space coordinates. The problem is relevant e.g. for description of the ultrafast dynamics of confined carriers. Particularly we consider a quantum wire, where the carriers are confined in the plane normal to the wire by infinite potentials. The initial condition is assumed both in energy and space coordinates. Grid implementations have been used to obtain the space and energy dependence of the evolution in the zero field case (see Figure 3).

SALUTE integrates a set of Monte Carlo algorithms for simulation of ultrafast carrier transport in semiconductors considered in above mentioned versions of the Wigner equation. A detailed description of the algorithms can be found in [20, 21, 22].

3 Grid Implementation

SALUTE solves an NP-hard problem concerning the evolution time using Monte Carlo algorithms which are inherently parallel. Thus, SALUTE is a very good candidate for implementations on MPI-enabled Grid sites. It is proved [20, 22] that stochastic error has order $O(N^{-1/2})$, where N is the number of samples of the MC estimator for a given evolution time t . When t is fixed and $N \rightarrow \infty$ the error decreases, but for N fixed and t large, the factor for the error looks ominous. That is why the problem of estimating the electron energy distribution function for long evolution times with small stochastic error requires combining both MC variance reduction techniques and distributed or parallel computations. By using the Grid environment provided by the EGEE project middleware, we were able to reduce the computing time of Monte Carlo simulations of ultra-fast carrier transport in semiconductors.

The numerical results discussed in the Figures 1-3 are obtained for zero temperature and GaAs material parameters: the electron effective mass is 0.063, the optimal phonon energy is $36meV$, the static and optical dielectric constants are $\epsilon_s = 10.92$ and $\epsilon_\infty = 12.9$. The initial condition at $t = 0$ is given by a function which is Gaussian in energy, scaled in a way to ensure, that the peak value is equal to unity [22].

Successful tests of the application were performed at the Bulgarian EGEE-GRID sites using the South Eastern Europe Resource Broker at the IPP-BAS.

Table 1. CPU times for SALUTE execution on MPI-enabled Grid site

Number of CPU	2	4	6
Time	9790	4896	3262

The MPI implementation was MPICH 1.2.6, and the execution is controlled from the Computing Element via the Torque batch system. The timing results for evolution time $t = 100$ femtoseconds are shown in Table 1. The parallel efficiency can clearly be seen from the results - with 4 CPUs, the running time is half less than the running time with 2 CPUs, with 6 CPUs the running time is 3 times less than with 2 CPUs.

4 Conclusion and Future Work

A Grid application that solves important problems in the field of nanotechnologies has been developed, studied and tested on MPI-enabled Grid sites. The test results show excellent parallel efficiency. Obtaining results for larger evolution times requires more computational power, which means that the application should run on larger sites or on several sites in parallel. The application can provide results for other types of semiconductors like Si or for composite materials.

The SALUTE application is able to fully utilize the computational power of Grids.

References

1. KALOS, M.A. AND P.A. WHITLOCK: *Monte Carlo Methods*. Volume 1: Basics, Wiley, New York, 1986.
2. KARAIVANOVA, A.: *Monte Carlo Methods for Parallel Processing*. PhD dissertation, IPP-BAS, 1996.
3. L. SMITH AND M. BULL: Development of mixed mode MPI/OpenMP applications. *Scientific Programming*, series 9 (2/3), 2001, 83–98.
4. B. R. BROOKS, R. E. BRUCCOLERI, B. D. OLAFSON, D. J. STATES, S. SWAMINATHAN, AND M. KARPLUS: CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamic Calculations. *J. Comp. Chem.*, **4**, 1893, 187–217.
5. J. A. RATHKOPF: *PMC: A shared short-cut to portable parallel power*. Lawrence Livermore National Laboratory, Livermore, CA, UCRL-112311, 1992.
6. J. FOSTER AND C. KESSELMANN: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
7. J. FOSTER, C. KESSELMANN AND S. TUECKE: The Anatomy of the Grid. *International Journal of Supercomputer Applications*, **15(3)**, 2001.
8. Globus website, <http://www.globus.org>.
9. Entropia: *PC Grid Computing*. <http://www.entropia.com>, 2002.
10. F.SOLMS AND W.H.STEEB: Distributed Monte Carlo integration using CORBA and Java. *International Journal of Modern Physics*, **9(7)**, 1998: 903-915.
11. M. H. ZHOU: *A Scientific Computing Tool for Parallel Monte Carlo in a Distributed Environment*. PhD dissertation, School of Mathematics, University of Southern Mississippi, 2000.
12. M. MASCAGNI, D. SEPERLEY AND A. SRINIVASAN: SPRNG: A Scalable Library for Pseudorandom Number Generation. *ACM Transactions on Mathematical Software*, 2000.
13. SPRNG website, <http://sprng.cs.fsu.edu>.

14. R. BUYYA, S. CHAPIN, AND D. DINUCCI: Architectural Models for Resource Management in the Grid. In: R. Buyya, M. Baker (Eds.): *Grid Computing*, GRID 2000, LNCS, vol. **1971**, Springer Verlag, 2000: 18–35.
15. R. ROSSI, C. JACOBONI, M. NEDJALOV: A Monte Carlo Solution of the Wigner transport Equation. *Semiconductor Science Technology*, **9**, 1994: 934-940.
16. M. NEDJALOV, I. DIMOV, F. ROSSI, C. JACOBONI. Convergent of the Monte Carlo Algorithms for the Solution of the Wigner Quantum-Transp. Equation. *J. Math. Comp. Mod.*, Vol. **23** (8/9), 1996: 159–166.
17. J. RAMMER: Quantum transport theory of electrons in solids: A single-particle approach. *Reviews of Modern Physics*, series **63(4)**, 1991, 781–817.
18. M. NEDJALOV, R. KOSIK, H. KOSINA, AND S. SELBERHERR: A Wigner Equation for Nanometer and Femtosecond Transport Regime. In: *Proceedings of the 2001 First IEEE Conference on Nanotechnology*, IEEE, 2001.
19. J. BARKER AND D. FERRY: Self-scattering path-variable formulation of high field time-dependent quantum kinetic equations for semiconductor transport in the finite-collision-duration regime. *Physical Review Letters*, series **42 (26)**, 1979: 1779–1781.
20. T.V. GUROV, M. NEDJALOV, P.A. WHITLOCK, H. KOSINA AND S. SELBERHERR: Femtosecond relaxation of hot electrons by phonon emission in presence of electric field. *Physica B*, **314**, 2002: 301–304.
21. T.V. GUROV AND I.T. DIMOV: A Parallel Monte Carlo Method for Electron Quantum Kinetic Equation. *LNCS*, **2907**, Springer-Verlang, 2004: 153–160.
22. T.V. GUROV, P.A. WHITLOCK: An efficient backward Monte Carlo estimator for solving of a quantum kinetic equation with memory kernel. *Mathematics and Computers in Simulation*, **60**, 2000: 85–105.

A Monte Carlo Algorithm for State and Parameter Estimation of Extended Targets*

Donka Angelova¹ and Lyudmila Mihaylova²

¹ Institute for Parallel Processing, Bulgarian Academy of Sciences
25A Acad. G. Bonchev St, 1113 Sofia, Bulgaria
donka@bas.bg

² Department of Communication Systems, Lancaster University,
South Drive, Lancaster LA1 4WA, UK
mila.mihaylova@ieee.org

Abstract. This paper considers the joint state and parameter estimation of extended targets. Both the target kinematic states, position and speed, are estimated with the target extent parameters. The developed algorithm is applied to a ship, whose shape is modelled by an ellipse. A Bayesian sampling algorithm with finite mixtures is proposed for the evaluation of the extent parameters whereas a suboptimal Bayesian interacting multiple model (IMM) filter estimates the kinematic parameters of the maneuvering ship. The algorithm performance is evaluated by Monte Carlo comparison with a particle filtering approach.

1 Introduction

The increasing interest in simulation-based Bayesian methods for analysis of dynamic models has been resulted in a variety of powerful techniques for filtering and prediction of complex dynamic systems [1, 2]. The problem of state and parameter estimation of dynamic systems has many applications such as in signal processing, machine learning, robotics, target (e.g., aircraft, ship) tracking [3, 4]. In the present work, a Monte Carlo (MC) algorithm is applied to ship size evaluation in the framework of state filtering (tracking) of a maneuvering ship, modeled by a Markovian jump system.

Most of the target tracking algorithms available in the literature consider the moving object as a single point and estimate its state vector based on the incoming sensor data, e.g. range and bearing. However, recent sensor systems are able to resolve individual features or measurement sources on an extended object. The possibility to additionally make use of this measurements is referred to extended target tracking. For example, a high-resolution radar can provide a measure of down-range object extent given a reasonable signal-to-noise ratio [5, 6]. This valuable information can help for more precise estimation of the object behaviour. It can assist in resolving measurement association uncertainty in situations of closely spaced objects in dense clutter. Furthermore, the knowledge of objects size is especially important for the purposes of classification.

* Research supported in part by the Bulgarian Foundation for Scientific Investigations: I-1202/02, I-1205/02, MI-1506/05 and by Center of Excellence BIS21++, 016639.

In this paper we address the problem of extended target tracking. There exist several ways for modelling object extent parameters. Models, based on measurements of individual points on object require complicated techniques for multiple hypotheses testing [7]. A simple ellipsoidal object model is proposed in [5] and adopted in our work. The lengths of the major and minor axes of the ellipse have to be calculated, based on the measurements of down-range extent. Shape parameters are included in [5] in the state vector together with kinematic parameters and are estimated by Extended and Unscented Kalman Filters (EKFs and UKFs) and particle filtering. However, it is pointed out in [5] that the EKF implementation is prone to divergence due to high nonlinearity conditions and a straightforward particle filter can avoid this problem.

Having in mind the inferences in [5, 6], we develop a Bayesian algorithm able to deal with the nonlinear estimation problem. Taking into account the possibility for subsequent classification, we assume that the unknown shape parameters are defined over the discrete set of values, with a given prior distribution. Within this formulation of the problem, *data augmentation* (DA) algorithm for finite mixture estimation [9] offers an alternative solution. DA represents a special case of Gibbs sampling [2] and belongs to the class of *Markov Chain Monte Carlo* (MCMC) methods. We develop a DA procedure for the parameter estimation, along with an IMM algorithm for kinematic state estimation. The scheme implemented here is inspired by the ideas in [8].

The paper is organised as follows. Section 2 describes the system dynamics and measurement model. Section 3 presents the formulation of the problem. The designed DA algorithm is presented in Section 4. Section 5 illustrates and compares the performance of the proposed algorithm with a particle filter (PF). Conclusions are given in Section 6.

2 System Dynamics and Measurement Models

Target motion tracking is performed by a recursive reconstruction of the state probability density function given the available prior information and current measurement data. Prior information includes dynamic models, models of the measurement process, initial state and noise probability distributions.

System Model. Consider the following model of a discrete-time jump Markov system, describing object dynamics and sensor measurements

$$\mathbf{x}_k = \mathbf{f}(m_k, \mathbf{x}_{k-1}, \boldsymbol{\theta}) + \mathbf{g}(m_k, \boldsymbol{\theta}) \mathbf{w}_k, \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}(m_k, \mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{d}(m_k, \boldsymbol{\theta}) \mathbf{v}_k, \quad k = 1, 2, \dots, \quad (2)$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the *base (continuous) state* vector, with transition function \mathbf{f} , $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is the measurement vector with measurement function \mathbf{h} , and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is a vector, containing unknown static parameters. The noises \mathbf{w}_k and \mathbf{v}_k are *independent identically distributed (i.i.d.)* Gaussian processes having characteristics $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q})$ and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$, respectively. All vectors and matrices are assumed of appropriate dimensions. The *modal (discrete) state* $m_k \in \mathbb{S} \triangleq \{1, 2, \dots, s\}$ is a first-order Markov chain with transition probabilities $p_{ij} \triangleq Pr\{m_k = j \mid m_{k-1} = i\}$, ($i, j \in \mathbb{S}$) and initial probability distribution $P_0(i) \triangleq Pr\{m_0 = i\}$, $i \in \mathbb{S}$, such that $P_0(i) \geq 0$, and $\sum_{i=1}^s P_0(i) = 1$. $k = 1, 2, \dots$ is a discrete time.

Consider a base state vector in the form $\mathbf{x}_k = (x_k, \dot{x}_k, y_k, \dot{y}_k)'$, where x and y specify the ship position with respect to an observer position, assumed known, and (\dot{x}, \dot{y}) is the velocity in the Cartesian plane, centered at the observer location. All possible s motion regimes of the maneuvering ship are modelled by the modal state variable m . The static parameter vector $\boldsymbol{\theta} = (\ell, \gamma)'$ contains shape parameters: the length of the major axis of the ship ellipse ℓ and the ratio of the lengths of the minor and major axes γ .

Measurement Equation. Similarly to [5, 6], we assume, that a high-resolution radar provides measurements of range r and bearing β to the object centroid, as well as the object down-range extent L along the observer-object line-of-sight (LOS). The relationship between L and the angle ϕ between the major axis of the ellipse and the target-observer LOS is given by

$$L(\phi) = \ell \sqrt{\cos^2 \phi + \gamma^2 \sin^2(\phi)}. \tag{3}$$

The measurement function \mathbf{h} in (2) is nonlinear,

$$\mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) = \begin{pmatrix} \sqrt{(x_k - x_o)^2 + (y_k - y_o)^2} \\ \arctan((y_k - y_o)/(x_k - x_o)) \\ L(\phi(\mathbf{x}_k)) \end{pmatrix} \tag{4}$$

where the measurement vector is $\mathbf{z}_k = (r_k, \beta_k, L_k)'$. Here (x_o, y_o) is the location of the observer. If it is assumed that the target ellipse is oriented so that its major axis is parallel to the velocity vector (\dot{x}, \dot{y}) then from (3) the along-range target extent can be written in terms of the state vector and $\boldsymbol{\theta}$ as

$$L(\phi(\mathbf{x}_k)) = \boldsymbol{\theta}(1) \sqrt{\cos^2 \phi(\mathbf{x}_k) + \boldsymbol{\theta}(2)^2 \sin^2 \phi(\mathbf{x}_k)}, \tag{5}$$

where $\phi(\mathbf{x}_k) = \arctan((x_k \dot{y}_k - \dot{x}_k y_k) / (x_k \dot{x}_k + y_k \dot{y}_k))$.

The problem that we consider has own particularities: the measurements of L are not used for the base state vector estimation. The kinematic states are estimated through r and β . The estimated kinematic states are, however, used for the estimation of ℓ and γ . This is the motivation for applying *separate estimators* to the two estimation problems.

3 Problem Formulation

The *goal* is to estimate the *state* vector \mathbf{x}_k and the *extent parameter* vector $\boldsymbol{\theta}$, based on all available measurement information $\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$. If we can calculate the *posterior joint state-size probability density function* (PDF)

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{Z}^k) = p(\boldsymbol{\theta} | \mathbf{x}_k, \mathbf{Z}^k) p(\mathbf{x}_k | \mathbf{Z}^k), \tag{6}$$

then for any integrable function $\bar{h}(\mathbf{x}_k, \boldsymbol{\theta})$ the required estimate is given by

$$E \{ \bar{h}(\mathbf{x}_k, \boldsymbol{\theta}) | \mathbf{Z}^k \} = \int \int \bar{h}(\mathbf{x}_k, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{x}_k, \mathbf{Z}^k) p(\mathbf{x}_k | \mathbf{Z}^k) d\boldsymbol{\theta} d\mathbf{x}_k. \tag{7}$$

If we denote the l -th mode history, realised by a Markovian jump system through time k as $m_k^l, l = 1, \dots, \varepsilon_k$ then the conditional PDF of the state is obtained as a Gaussian mixture with an exponentially increasing number of terms [10]

$$p(\mathbf{x}_k | \mathbf{Z}^k) = \sum_{l=1}^{\varepsilon_k} p(\mathbf{x}_k | m_k^l, \mathbf{Z}^k) P(m_k^l | \mathbf{Z}^k). \quad (8)$$

The exponentially increasing computations can be avoided by different ways of combining histories of models. For example, the Interacting Multiple Model (IMM) filter [10] provides an approximate state estimate $\hat{\mathbf{x}}_k = E\{\hat{h}(\mathbf{x}_k) | \mathbf{Z}^k\}$ and its associated covariance matrix \mathbf{P}_k , by using s working in parallel EKFs. Then the estimate of $\boldsymbol{\theta}$ can be expressed as $\hat{\boldsymbol{\theta}}_k = E\{\hat{h}(\boldsymbol{\theta}_k) | \hat{\mathbf{x}}_k, \mathbf{Z}^k\}$. Note that the k index of $\boldsymbol{\theta}$ indicates that it is calculated based on the information up to time instant k , not that $\boldsymbol{\theta}$ is time-varying.

MC algorithms (PF or MCMC) can be applied to the highly nonlinear extent estimation problem. The use PF for the first step is not justifiable, as ships exhibit moderate maneuvers and the time interval between the measurements is rather short. Since the size estimate could be obtained along with the filtering process, but not necessary on-line, the more precise MCMC algorithm can be applied.

The scheme implemented here is similar to [8]. It comprises the following steps. On receipt of a new measurement \mathbf{z}_k :

- a) run the IMM algorithm with the previous state estimate $\hat{\mathbf{x}}_{k-1}$ in order to update the current state estimate $\hat{\mathbf{x}}_k$.
- b) find the estimate $\hat{\boldsymbol{\theta}}_k$ of the parameter vector $\boldsymbol{\theta}$ based on the previous estimates $\hat{\boldsymbol{\theta}}_{k-1}$, $\hat{\mathbf{x}}_k$, and the measurement likelihood $p(\mathbf{z}_k | \hat{\boldsymbol{\theta}}_{k-1}, \mathbf{z}_{k-1})$, by PF or DA scheme.

4 Extent Parameters Estimation by Stochastic Simulation

Based on a priori information about ship types, we assume that $\boldsymbol{\theta}$ takes values from a known discrete set $\boldsymbol{\theta} \in \mathbb{T} \triangleq \{1, 2, \dots, t\}$ with known prior distribution: $P_{\boldsymbol{\theta}_0}(i) \triangleq Pr\{\boldsymbol{\theta} = i\}, i \in T$, such that $P_{\boldsymbol{\theta}_0}(i) \geq 0$, and $\sum_{i=1}^t P_{\boldsymbol{\theta}_0}(i) = 1$. Let us suppose that along-range extent measurements $L_k, k = 1, \dots, n, \dots$ have Gaussian distributed zero-mean errors with known variance R_L . The PDF of the measurement L_k is represented in the following mixture form [9, 11]:

$$p(L_k | \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{x}_k) = \sum_{j=1}^t \pi_j G(L_k | \boldsymbol{\theta}_j, \mathbf{x}_k), \quad (9)$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_t)$ are the mixture proportions which are constrained to be non-negative and sum to unity. $G(L_k | \boldsymbol{\theta}_j, \mathbf{x}_k) \sim N(L_k; L(\boldsymbol{\theta}_j, \hat{\mathbf{x}}_k), R_L)$ is a Gaussian density and $L(\boldsymbol{\theta}_j, \hat{\mathbf{x}}_k)$ is the measurement prediction, calculated according to (5). Thus, the task is reduced to the well known *finite mixture estimation problem*: for the mixture model (9) with *known component* PDFs $G(L_k | \boldsymbol{\theta}_j, \mathbf{x}_k)$, one needs to estimate the *unknown weights* $\boldsymbol{\pi} = (\pi_1, \dots, \pi_t)$, given a sequence of independent observations L_1, L_2, \dots, L_n . The mixture component with a *maximum weight* identifies the

most probable ship type. The estimate of the extent parameters can be calculated as the weighted by π sum of the possible θ values in the set.

Mixture Weights Estimation by Data Augmentation. DA algorithm approximately evaluates the mixture posterior distribution, relying on the missing data structure of mixture model. Generally, the mixture model is given by the observation of n independent random variables y_1, \dots, y_n from a t -component mixture [9, 11],

$$F(y_k) = \sum_{j=1}^t \pi_j F_j(y_k), \quad k = 1, \dots, n, \tag{10}$$

where the densities $F_j, j = 1, \dots, t$ are known or are known up to a parameter and the proportions π_j satisfy the above conditions. We consider the special case, where *only* the weights π have to be estimated. According to [9], the mixture model can always be expressed in terms of missing (or incomplete) data. That is, define vectors $\delta(k) = (\delta_1(k), \delta_2(k), \dots, \delta_t(k)), k = 1, 2, \dots, n$ with components $\delta_j(k) \in \{0, 1\}, j = 1, 2, \dots, t$, which indicate that the measurement y_k has density $F_j(y_k)$ [8]. The model is hierarchical with, on top, the true parameters of the mixture, π , then the missing data whose distribution depends on $\pi, \delta \sim p(\delta|\pi)$, and, at the bottom, the observed data $\mathbf{y} \sim p(\mathbf{y}|\pi, \delta)$. Starting with an initial value $\pi^{(0)}$, the algorithm implements *two-step iterative scheme*: at the iteration $u, u = 1, 2, \dots$

- a) generate $\delta^{(u)} \sim p(\delta|\mathbf{y}, \pi^{(u)})$ from a multinomial distribution with weights proportional to the observation likelihoods: $\delta_j^{(u)}(k) \propto \pi_j^{(u)} F_j(y_k)$;
- b) generate $\pi^{(u+1)} \sim p(\pi|\mathbf{y}, \delta^{(u)})$.

Since the conjugate priors on π are with Dirichlet distributions (DD) $\mathcal{D}(\alpha_1, \dots, \alpha_t)$ [9], $\pi^{(u+1)}$ is generated according to the DDs with parameters, depending on the missing data. Bayesian sampling produces an ergodic Markov chain $(\pi^{(u)})$ with stationary distribution $p(\pi|\mathbf{y})$. Thus, after u_0 initial (warming up) steps, a set of U samples $\pi^{(u_0+1)}, \dots, \pi^{(u_0+U)}$ are approximately distributed as $p(\pi|\mathbf{y})$ and, due to ergodicity, averaging can be made with respect to time [8, 9].

In the next scheme, the sequence of observations $y_k, k = 1, \dots$ is replaced by the along-range extent measurements $L_k, k = 1, \dots$. The joint IMM and data augmentation scheme is given below. DA is realised in a sliding window mode.

Algorithm Outline

For $k = 1, 2, \dots$

- Run the IMM algorithm with the previous state vector $\hat{\mathbf{x}}_{k-1}$, covariance matrix \mathbf{P}_{k-1} and posterior model probability vector $\boldsymbol{\mu}_{k-1}$ to update the current estimate $\hat{\mathbf{x}}_k$, together with \mathbf{P}_k and $\boldsymbol{\mu}_k$.
- Compute Mixture Components Conditional PDFs

$$\tilde{G}_j(k) \triangleq G(L_k|\boldsymbol{\theta}_j, \mathbf{x}_k) \propto \exp \left[-0.5 (L_k - L(\boldsymbol{\theta}_j, \hat{\mathbf{x}}_k))^T R_L^{-1} (L_k - L(\boldsymbol{\theta}_j, \hat{\mathbf{x}}_k)) \right]$$

- Implement Data Augmentation
 - Initialisation: $\pi^{(0)} = \pi(k-1)$

- Iterations ($u = 0, 1, \dots, u_0 + U - 1$)
 - * Missing Data Conditional Probability Mass Functions (PMFs)

$$q_j^{(u)}(l) = \frac{\pi_j^{(u)} \tilde{G}_j(l)}{\sum_{j=1}^t \pi_j^{(u)} \tilde{G}_j(l)}, \quad l = 1, 2, \dots, k, \quad j = 1, 2, \dots, t$$

- * Missing Data Generation (Multinomial Sampling)

$$\delta^{(u)}(l) = (0, \dots, 0, 1, 0, \dots, 0) \sim \left\{ q_j^{(u)}(l) \right\}_{j=1}^t, \quad l = 1, 2, \dots, k$$

- * Parameter Evaluation (Dirichlet Distribution Sampling)

$$\pi^{(u+1)} \sim \mathcal{D} \left(\pi; \alpha_1 + \sum_{l=1}^k \delta_1^{(u)}(l), \dots, \alpha_t + \sum_{l=1}^k \delta_t^{(u)}(l) \right)$$

- Calculate the Output Estimates

$$\pi(k) = \frac{1}{U} \sum_{\sigma=1}^U \pi^{(u_0+\sigma)} \quad \text{and} \quad \hat{\theta}_k = \sum_{j=1}^t \pi_j(k) \theta_j$$

5 Simulation Results

The algorithm performance is evaluated by simulations over trajectories, including consecutive segments of uniform motion and maneuvers (a typical scenario is shown in Fig.1(a)). The observer is static, located at the origin of $x - y$ plane. The initial target state is $\mathbf{x}_0 = (18e3, -14, 90e3, 5)'$. It performs two turn maneuvers with a normal acceleration of $\pm 1.4 [m/s^2]$. The object length is $\ell = 50 [m]$ and the ratio of the lengths of the minor and major axes (aspect ratio) is $\gamma = 0.2$. The sensor parameters are as follows [6]: sampling interval $T = 0.2 [s]$; the standard deviations of measurement errors along range, azimuth and along-range target extent are respectively:

$$\sigma_r = 5 [m], \sigma_\beta = 0.2 \text{ deg} \text{ and } \sigma_L = 5 [m].$$

Root-Mean Squared Errors (RMSEs) [10] are selected as a quantitative measure for the algorithm performance evaluation.

An IMM algorithm with $s = 3$ models is designed. The first model corresponds to the nearly constant velocity motion. The next two models are matched to the nearly coordinated turn maneuvers with a turn rate of $\omega = \pm 1.4 [rad/s]$. The form of the transition matrices in (1) for these models can be found in [10].

A particle filter for extent parameters estimation is realised for the purposes of comparison with DA algorithm, with $N = 300$ particles. Similarly to the procedure, described in [1] (Ch.10), N particles $(\theta^{(i)})_{i=1}^N$ are generated according to a priori normal distribution with mean, corresponding to the true θ . After that the particles are predicted according to a normal distribution with small deviations to provide “artificial

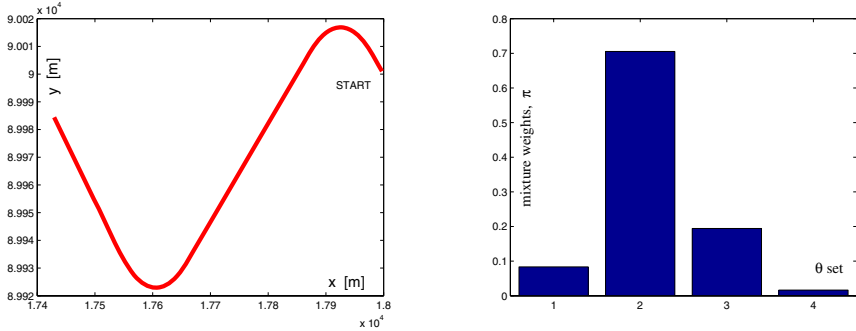


Fig. 1. Object trajectory and Mixture proportions π

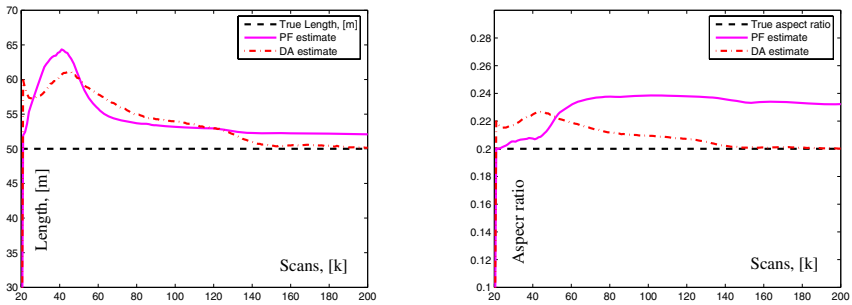


Fig. 2. PF and DA comparison: True and estimated ℓ and γ

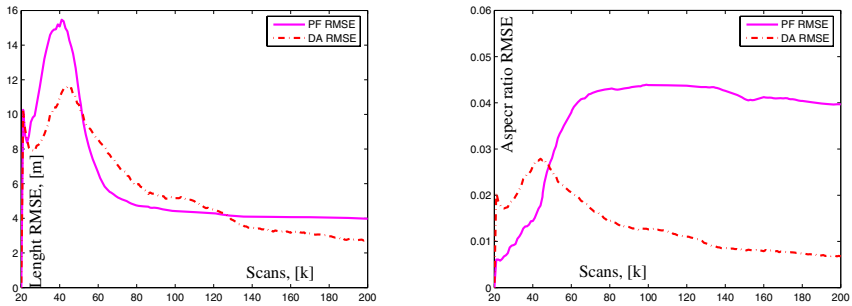


Fig. 3. PF and DA comparison: RMSE of ℓ and γ

evolution” of parameters. Then the particle weights are evaluated using likelihoods of the received measurements and finally resampling is implemented according to a known rule.

In accordance with (Sec. 4), we assume that θ takes values from a discrete set of $t = 4$ components: $\{(30, 0.15), (50, 0.2), (70, 0.25), (100, 0.3)\}$ with equal initial probabilities. θ_2 corresponds to the true θ . The mixture proportions $\pi_j, j = 1, \dots, t$, estimated by the DA procedure are given in Fig.1(b). It can be seen that DA identifies

the correct θ with a high probability. The results are obtained from one MC realisation and confirm the reliability of the algorithm for classification tasks.

Figs.(2) and (3) demonstrate the better performance of the DA in comparison with the PF: the estimation errors of ℓ and γ obtained by the PF are quite large. It is natural to expect a better performance of DA scheme. It has additional prior information, available from the θ set. DA is an off-line procedure and processes the cumulative measurement information. In addition, PF involves additional noise, necessary for prediction which deteriorates the estimation accuracy of the parameters (they are fixed). Its performance is highly sensitive to the choice of noise deviations, which are design parameters. However, the better performance is achieved at the increased computational time. The relative execution time is approximately 17:1 in PF favour.

6 Conclusion

An alternative solution to the problem of extended object tracking is proposed in this paper. Two different Bayesian algorithms are developed for kinematic state and size parameter estimation of a ship, based on positional and along-range object extent measurements. A Monte Carlo algorithm (Data augmentation) is designed for length and aspect ratio assessment of the elliptical ship shape. The performance of the proposed algorithm is evaluated by Monte Carlo simulation. The results show that DA is capable to deal with highly nonlinear relationships between states, parameters, measurements and complicated target-observer geometry. A combined IMM-DA procedure is proposed which provides accurate joint state-parameter estimation and reliable identification of the ship type.

References

1. Doucet, A., de Freitas, N., Gordon, N. (ed.): *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, (2001).
2. Liu, J.: *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, (2003).
3. Storvik, G.: Particle filters in state space models with the presence of unknown static parameters. *IEEE. Trans. of Signal Processing*, Vol. **50**, No. 2, (2002) 281–289.
4. Doucet, A., Tadic, V.: Parameter estimation in general state-space models using particle methods. *Ann. Inst. Stat. Math.*, vol. **55**, No. 2, (2003) 409–422.
5. Salmond, D., Parr, M.: Track maintenance using measurements of target extent. *IEE Proc.-Radar Sonar Navig.*, Vol. **150**, No. 6, (2003) 389–395.
6. Ristic, B., Salmond, D.: A study of a nonlinear filtering problem for tracking an extended target. *Proc. Seventh Intl. Conf. on Information Fusion*, (2004) 503–509.
7. Vermaak, J., N. Ikoma, S. Godsill: Sequential Monte Carlo Framework for Extended Object Tracking. *IEE Proc.-Radar Sonar Navig.*, Vol. **152**, No. 5 (2005), 353–363.
8. Jilkov, V., Li, X. Rong, Angelova, D.: Estimation of Markovian Jump Systems with Unknown Transition Probabilities through Bayesian Sampling. *LNCS*, Vol. 2542 (2003), Springer-Verlag, Berlin, pp. 307–315.
9. Diebolt, J., Robert, C.P.: Estimation of Finite Mixture Distributions through Bayesian Sampling. *J. of Royal Statist. Soc. B* **56**, No. 4, (1994) 363–375.
10. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. Wiley, New York (2001).
11. Stephens, M.: *Bayesian methods for mixture of normal distributions*. PhD Thesis, 1997.

Error Analysis of a Monte Carlo Algorithm for Computing Bilinear Forms of Matrix Powers*

Ivan Dimov, Vassil Alexandrov, Simon Branford, and Christian Weihrauch

Centre for Advanced Computing and Emerging Technologies
School of Systems Engineering, The University of Reading
Whiteknights, PO Box 225, Reading, RG6 6AY, UK

{v.n.alexandrov, s.j.branford, i.t.dimov, c.weihrauch}@reading.ac.uk

Institute for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bonchev 25 A, 1113 Sofia, Bulgaria
ivdimov@bas.bg

Abstract. In this paper we present error analysis for a Monte Carlo algorithm for evaluating bilinear forms of matrix powers. An almost Optimal Monte Carlo (MAO) algorithm for solving this problem is formulated. Results for the structure of the probability error are presented and the construction of robust and interpolation Monte Carlo algorithms are discussed.

Results are presented comparing the performance of the Monte Carlo algorithm with that of a corresponding deterministic algorithm. The two algorithms are tested on a well balanced matrix and then the effects of perturbing this matrix, by small and large amounts, is studied.

Keywords: Monte Carlo algorithms, matrix computations, performance analysis, iterative process.

1 Introduction

The problems of inverting a real $n \times n$ matrix (MI), solving a system of linear algebraic equations (SLAE) or finding extreme eigenvalues are of unquestionable importance in many scientific and engineering applications: e.g real-time speech coding, digital signal processing, communications, stochastic modelling, and many physical problems involving partial differential equations. The direct methods of solution for SLAE require $O(n^3)$ sequential steps when using the usual elimination or annihilation schemes (e.g. Gaussian elimination, Gauss-Jordan methods) [7] and, similarly, the direct methods for MI or calculating extreme eigenvalues can be computationally expensive. Consequently the computation time for very large problems, or for real-time solution problems, can be prohibitive and this prevents the use of many established algorithms.

It is known that Monte Carlo methods give statistical estimates for elements of the inverse matrix, or for components of the solution vector of SLAE, by

* Partially supported by the Bulgarian Ministry of Education and Science, under grant I-1405/2004.

performing random sampling of a certain random variable, whose mathematical expectation is the desired solution [8, 9]. The problem of variance estimation, in the optimal case, has been considered for extremal eigenvalues [10]. In this paper we extend this by considering bilinear forms of matrix powers, which can be used to formulate solutions for all three problems, and study the effects, on the Monte Carlo algorithm, of perturbing a well balanced matrix.

The idea of Monte Carlo and an algorithm for the problem of bilinear forms of matrix powers is presented in Section 2; in Section 3 we detail the implementation we used for the algorithm; the results of the experimental runs of this implementation are also presented in this section; and we conclude the work in Section 4.

2 Formulation of the Monte Carlo Algorithm

2.1 Bilinear Forms of Matrix Powers

We are interested in the bilinear form of matrix powers:

$$(v, A^k h). \tag{1}$$

For x , the solution of a SLAE $Bx = b$ then

$$(v, x) = \left(v, \sum_{i=0}^k A^i h \right), \tag{2}$$

where the Jacobi Over-relaxation Iterative Method has been used to transform the SLAE into the problem $x = Ax + h$. In cases where the Neumann series does not converge a resolvent method can be used [5, 6].

Matrix inversion is equivalent to solving (2) n times $Bc_j = e^{(j)}$, $j = 1, \dots, n$ for the special case where $c_j \equiv (c_{1j}, \dots, c_{nj})^T$ and $e^{(j)} \equiv (0, \dots, 0, \underbrace{1}_j, 0, \dots, 0)^T$.

For an arbitrary large natural number k the Rayleigh quotient can be used to obtain an approximation for λ_1 , the dominant eigenvalue, of a matrix A :

$$\lambda_1 \approx \frac{(v, A^k h)}{(v, A^{k-1} h)}.$$

Thus it is clear that having an efficient way of calculating (1) is important. This is especially important in cases where we are dealing with large and/or sparse matrices.

2.2 Almost Optimal Markov Chain Monte Carlo

We shall use the so-called Almost Optimal Monte Carlo (MAO) algorithm studied in [2, 3, 1, 5, 4]. Here we give a brief presentation of MAO.

Suppose we have a Markov chain

$$T = \alpha_0 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \dots \rightarrow \alpha_k \rightarrow \dots$$

with n states. The random trajectory (chain) T_k of length k starting in the state α_0 is defined as follows:

$$T_k = \alpha_0 \rightarrow \alpha_1 \rightarrow \dots \rightarrow \alpha_j \rightarrow \dots \rightarrow \alpha_k, \tag{3}$$

where α_j means the number of the state chosen, for $j = 1, \dots, k$.

Assume that

$$P(\alpha_0 = \alpha) = p_\alpha, \quad P(\alpha_j = \beta | \alpha_{j-1} = \alpha) = p_{\alpha\beta},$$

where p_α is the probability that the chain starts in state α and $p_{\alpha\beta}$ is the transition probability to state β after being in state α . Probabilities $p_{\alpha\beta}$ define a transition matrix P . We require that

$$\sum_{\alpha=1}^n p_\alpha = 1 \quad \text{and} \quad \sum_{\beta=1}^n p_{\alpha\beta} = 1, \quad \text{for any } \alpha = 1, 2, \dots, n. \tag{4}$$

We will consider a special choice of density distributions p_i and p_{ij} defined as follows:

$$p_i = \frac{|v_i|}{\|v\|}, \quad \|v\| = \sum_{i=1}^n |v_i| \quad \text{and} \quad p_{ij} = \frac{|a_{ij}|}{\|a_i\|}, \quad \|a_i\| = \sum_{j=1}^n |a_{ij}|. \tag{5}$$

2.3 Monte Carlo Algorithm for Computing Bilinear Forms of Matrix Powers $(v, A^k h)$

The pair of density distributions (5) defines a finite chain of vector and matrix entrances:

$$v_{\alpha_0} \rightarrow a_{\alpha_0\alpha_1} \rightarrow \dots \rightarrow a_{\alpha_{k-1}\alpha_k}. \tag{6}$$

The latter chain induces the following product of matrix/vector entrances and norms:

$$A_v^k = v_{\alpha_0} \prod_{s=1}^k a_{\alpha_{s-1}\alpha_s}; \quad \|A_v^k\| = \|v\| \times \prod_{s=1}^k \|a_{\alpha_{s-1}}\|.$$

Note, that the product of norms $\|A_v^k\|$ is not a norm of A_v^k . The rule for creating the value of $\|A_v^k\|$ is the following: the norm of the initial vector v , as well as norms of all row-vectors of matrix A visited by the chain (6), defined by densities (5), are included. For such a choice of densities p_i and p_{ij} we have

$$E\{h_{\alpha_k}\} = \frac{\text{sign}\{A_v^k\}}{\|A_v^k\|} (v, A^k h). \tag{7}$$

The standard deviation $\sigma\{h_{\alpha_k}\}$ is finite. Since random variable

$$\theta^{(k)} = \text{sign}\{A_v^k\} \times \|A_v^k\| h_{\alpha_k}$$

is an unbiased estimate of the form $(v, A^k h)$, (7) can be used to construct a MC algorithm.

Let us consider N realizations of the Markov chain T_k (3) defined by the pair of density distributions (5). Denote by $\theta_i^{(k)}$ the i^{th} realization of the random variable $\theta^{(k)}$. Then the value

$$\bar{\theta}^{(k)} = \sum_{i=1}^N \theta_i^{(k)} = \text{sign}\{A_v^k\} \| A_v^k \| \sum_{i=1}^N \{h_{\alpha_k}\}_i \tag{8}$$

can be considered as a MC approximation of the form $(v, A^k h)$. The probability error of this approximation can be presented in the following form:

$$R_N^{(k)} = \left| (v, A^k h) - \bar{\theta}^{(k)} \right| = c_p \sigma\{\theta^{(k)}\} N^{-\frac{1}{2}}, \tag{9}$$

where the constant c_p only depends on the probability $P = Pr\{|\bar{\theta} - J| \leq R_N\}$ (where J is the exact solution of the problem, R_N is the probability error and $0 < P < 1$) and does not depend on N and on $\theta^{(k)}$. Because of the finiteness of the standard deviation the probability error is always finite.

In fact, (8) together with the sampling rules using probabilities (5) defines a MC algorithm. The expression (8) gives a MC approximation of the form $(v, A^k h)$ with a probability error $R_N^{(k)}$. Obviously, the quality of the MC algorithm depends on the behaviour of the standard deviation $\sigma\{\theta^{(k)}\}$. So, there is a reason to consider a special class of *robust MC algorithms*.

2.4 Robust and Interpolation Monte Carlo Algorithms

Definition. A MC algorithm for which the standard deviation does not increase with the increasing of the matrix power k is called a *robust MC algorithm*.

Thus, if the MC algorithm is robust, then there exists a constant M such that

$$\lim_{k \rightarrow \infty} \sigma\{\theta^{(k)}\} \leq M.$$

Definition. A MC algorithm for which the probability error is zero is called an *interpolation MC algorithm*.

So, using the the following notations:

$$\hat{h} = \{h_i^2\}_{i=1}^n, \quad \bar{v} = \{|v_i|\}_{i=1}^n, \quad \bar{A} = \{|a_{ij}|\}_{i,j=1}^n$$

we can prove that

$$\sigma^2\{\theta^{(k)}\} = \| A_v^k \| \left(\bar{v}, \bar{A}^k \hat{h} \right) - (v, A^k h)^2,$$

where $\sigma^2\{\theta^{(k)}\}$ is the variance. A finite and small variance means that the associated Monte Carlo method should converge, and the probable error will be small.

Now we can formulate an important result that gives a sufficient condition for constructing an interpolation MC algorithm.

Let $h = (1, \dots, 1)$, $v = (\frac{1}{n}, \dots, \frac{1}{n})$ and

$$A = \begin{pmatrix} \frac{1}{n} & \dots & \frac{1}{n} \\ \vdots & & \vdots \\ \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}.$$

Then MC algorithm defined by density distributions (5) is an interpolation MC algorithm. Matrices A of the above form are *stochastic matrices*.

3 Experimental Results

For this paper two programs were implemented; one for the proposed Monte Carlo algorithm and one for a deterministic matrix-vector and vector-vector multiplication solving the same problem. The algorithms were written in Fortran 90 and compiled with the Intel Fortran 9.0 compiler. For all real numbers double precision was used to lower the influence of rounding errors. Random numbers were generated with the help of the Fortran 90 `RANDOM_NUMBER()` sub-routine which creates random numbers with a uniform distribution.

The results were generated on a SGI Prism equipped with 8 x 1.5 GHz Itanium II processors and 16 GByte of main memory. For the input data the vectors h and v and matrix A were generated for the sizes 100, 1000 and 5000. The vector h was filled with ones and v was filled with $\frac{1}{n}$. The matrix A was filled with elements of size $\frac{1}{n}$ and then perturbed by 2, 5, 10, 50 and 90%. The norm of such matrices is around 1. For comparison random non-balanced matrices were generated too. In Figure 1 we show the results for different perturbations for the Monte Carlo algorithm and for the deterministic code. We see that the results are very close for perturbations of up to 10% whereas the results for 50 and 90% differ up to 2% for matrices of size 1000 and 5000 and differ up to 14% for a matrix of size 100.

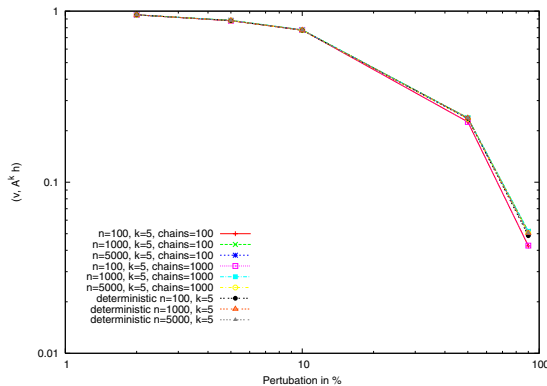


Fig. 1. The dependence of MC results on perturbation of A

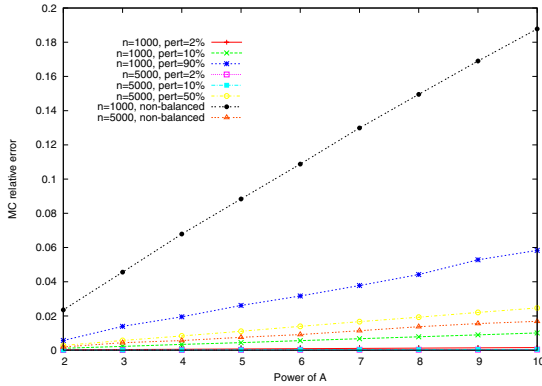


Fig. 2. The dependence of MC relative error on power of A

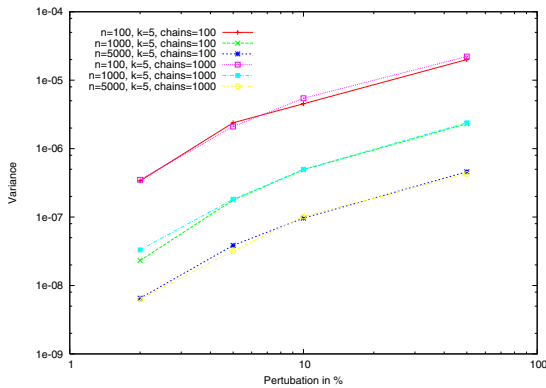


Fig. 3. The dependence of MC variance on perturbation of A

Since the deterministic computations were performed with a double precision we accept the results obtained as *exact results* and use them to analyse the accuracy of the results produced by our Monte Carlo code. In Figure 2 the relative error of the results for Monte Carlo algorithm is shown. The Monte Carlo relative error was computed by using the following formulas:

$$\text{MC error} = |\text{MC result} - \text{exact result}|,$$

$$\text{MC relative error} = \frac{\text{MC error}}{\text{exact result}}.$$

From Figure 2 we can see that the error increases linearly if k is increasing. The larger the matrix is, the smaller the influence of the perturbation. For comparison, the results for non-balanced matrices were included.

The variance of the results for the different perturbations are shown in Figure 3. In this figure we compare results for different sizes of the matrix and different chain lengths. Again it is obvious that the influence of the perturbation

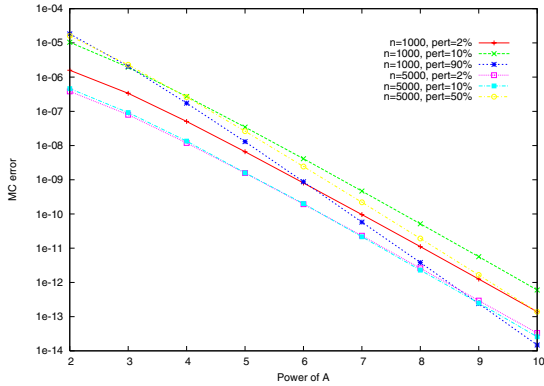


Fig. 4. The dependence of MC error on power of matrices with "small" spectral norms ($\|A\| \approx 0.1$)

is a lot bigger for smaller matrix of size 100. But over all a variance of $10e-06$ is a good result and shows that the Monte Carlo algorithm works well with this kind of balanced matrices.

In order to test the robustness of the Monte Carlo algorithm, a re-run of the experiments was done with matrices of norm smaller than 1. Therefore the randomly generated matrices were re-used and their elements were divided by 10. The results for these experiments are shown in Figure 4 and Figure 5.

In Figure 4 the Monte Carlo errors for matrix size of $n = 1000$ and chain length of 1000 are shown. We can see that the Monte Carlo algorithm is very robust because with an increasing k the error is decreasing enormously.

The variance shown in Figure 5 is 10^{10} smaller than the variance shown in Figure 3.

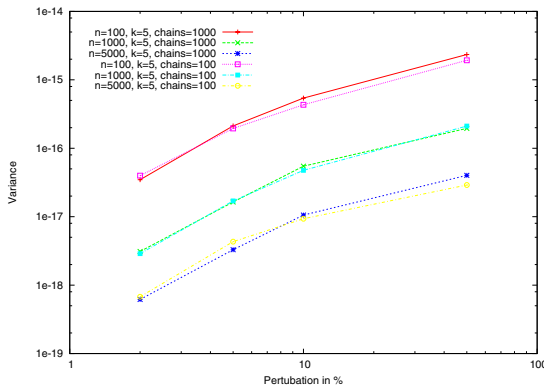


Fig. 5. The dependence of MC variance on perturbation of matrices with "small" spectral norms ($\|A\| \approx 0.1$)

4 Conclusion

In this paper we have analysed the error and robustness of the proposed MC algorithm for computing bilinear form of matrix powers $(v, A^k h)$. We have shown that with increasing the perturbations the error and the variance are increasing too. Especially small matrices have a high variance. For a rising power of A an increase of the relative error can be observed. The robustness of the Monte Carlo algorithm with balanced matrices with matrix norms much smaller than 1 has been demonstrated. In these cases the variance has improved a lot compared to cases were matrices have norms close to 1. We can conclude that the balancing of the input matrix is very important for MC computations. A balancing procedure should be performed as an initial (preprocessing) step in order to improve the quality of Monte Carlo algorithms. For matrices that are "close" in some sense to the stochastic matrices the accuracy of the MC algorithm is very high.

References

1. V. Alexandrov, E. Atanassov, I. Dimov, *Parallel Quasi-Monte Carlo Methods for Linear Algebra Problems*, Monte Carlo Methods and Applications, Vol. 10, No. 3-4 (2004), pp. 213-219.
2. I. Dimov, *Minimization of the Probable Error for Some Monte Carlo methods*. Proc. Int. Conf. on Mathematical Modeling and Scientific Computation, Albena, Bulgaria, Sofia, Publ. House of the Bulgarian Academy of Sciences, 1991, pp. 159-170.
3. I. Dimov, *Monte Carlo Algorithms for Linear Problems*, Pliska (Studia Mathematica Bulgarica), Vol. 13 (2000), pp. 57-77.
4. I. Dimov, T. Dimov, T. Gurov, *A New Iterative Monte Carlo Approach for Inverse Matrix Problem*, Journal of Computational and Applied Mathematics, Vol. 92 (1997), pp. 15-35.
5. I.T. Dimov, V. Alexandrov, *A New Highly Convergent Monte Carlo Method for Matrix Computations*, Mathematics and Computers in Simulation, Vol. 47 (1998), pp. 165-181.
6. I. Dimov, A. Karaivanova, *Parallel computations of eigenvalues based on a Monte Carlo approach*, Journal of Monte Carlo Method and Applications, Vol. 4, Nu. 1, (1998), pp. 33-52.
7. G.V. Golub, C.F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins Univ. Press, Baltimore, 1996.
8. I.M. Sobol *Monte Carlo numerical methods*, Nauka, Moscow, 1973.
9. J.R. Westlake, *A Handbook of Numerical matrix Inversion and Solution of Linear Equations*, John Wiley & Sons, inc., New York, London, Sydney, 1968.
10. M. Mascagni, A. Karaivanova, *A Parallel Quasi-Monte Carlo Method for Computing Extremal Eigenvalues*, Monte Carlo and Quasi-Monte Carlo Methods (2000), Springer, pp. 369-380.

Comparison of the Computational Cost of a Monte Carlo and Deterministic Algorithm for Computing Bilinear Forms of Matrix Powers

Christian Weihrauch, Ivan Dimov, Simon Branford, and Vassil Alexandrov

Centre for Advanced Computing and Emerging Technologies
School of System Engineering, The University of Reading
Whiteknights, PO Box 225, Reading, RG6 6AY, UK
{c.weihrauch, i.t.dimov, s.j.branford, v.n.alexandrov}@reading.ac.uk
Institute for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bonchev 25 A, 1113 Sofia, Bulgaria
ivdimov@bas.bg

Abstract. In this paper we consider bilinear forms of matrix polynomials and show that these polynomials can be used to construct solutions for the problems of solving systems of linear algebraic equations, matrix inversion and finding extremal eigenvalues. An almost Optimal Monte Carlo (MAO) algorithm for computing bilinear forms of matrix polynomials is presented.

Results for the computational costs of a balanced algorithm for computing the bilinear form of a matrix power is presented, i.e., an algorithm for which probability and systematic errors are of the same order, and this is compared with the computational cost for a corresponding deterministic method.

Keywords: Monte Carlo algorithms, matrix computations, performance analysis, computational cost, iterative process.

1 Introduction

Many scientific and engineering applications are based on the problems of finding extremal eigenvalues, solving a system of linear algebraic equations (SLAE), or inverting a real $n \times n$ matrix (MI). The computation time for very large problems, or for finding solutions in real-time, can be prohibitive and this prevents the use of many established algorithms. Monte Carlo methods give statistical estimates of the required solution, by performing random sampling of a random variable, whose mathematical expectation is the desired solution [10, 11].

Several authors have presented work on the estimation of computational complexity of linear algebra problems [4, 5, 6, 12, 13, 14]. In this paper we consider bilinear forms of matrix powers, which can be used to formulate solutions for all three problems. Considering the set, \mathcal{A} , of algorithms, A , for calculating bilinear forms of matrix powers

$$\mathcal{A} = \{A : Pr\{r_{n,N} \leq \varepsilon\} \geq c\},$$

with a probability error less than a given constant ε , there is the practical question of which algorithm in the set has the smallest computational cost. In this paper we compare the computational cost of two such method - a Monte Carlo algorithm and a deterministic algorithm.

The formulation of MI, SLAE and finding extreme eigenvalues in terms of bilinear forms of matrix powers is presented in Section 2; in Section 3 we present a Monte Carlo algorithm for finding the bilinear form of a matrix power; the computational cost of the Monte Carlo algorithm and of the deterministic method are presented in Section 4; and we conclude the work in Section 5.

2 Formulation of the Problems

In this paper we are interested in the evaluation of forms

$$(v, p(A)h), \tag{1}$$

where $p(A)$ is a matrix polynomial and $v, h \in \mathbb{R}^n$ are arbitrary vectors.

2.1 Bilinear Form of Matrix Powers

In a special case of $p(A) = A^k$ then (1) becomes

$$(v, A^k h).$$

2.2 Eigenvalues of Matrices

The well-known Power method [9] gives an estimate for the dominant eigenvalue λ_1 , of a matrix A . This estimate is called *Rayleigh quotient*:

$$\lambda_1 = \lim_{k \rightarrow \infty} \frac{(v, A^k h)}{(v, A^{k-1} h)},$$

where $v, h \in \mathbb{R}^n$ are arbitrary vectors. The Rayleigh quotient is used to obtain, for an arbitrary large natural number k , an approximation:

$$\lambda_1 \approx \frac{(v, A^k h)}{(v, A^{k-1} h)}. \tag{2}$$

To construct an algorithm for evaluating the minimal by modulo eigenvalue, λ_n , one has to consider the following matrix polynomial:

$$p(A) = \sum_{k=0}^{\infty} q^k C_{m+k-1}^k A^k, \tag{3}$$

where C_{m+k-1}^k are binomial coefficients, and the characteristic parameter, q , is used as acceleration parameter of the algorithm [4, 7, 8].

If $|qA| < 1$, then the polynomial (3) becomes the resolvent matrix [6, 7]:

$$p(A) = \sum_{k=0}^{\infty} q^k C_{m+k-1}^k A^k = [I - qA]^{-m} = R_q^m,$$

where $R_q = [I - qA]^{-1}$ is the resolvent matrix of the equation

$$x = qAx + h. \tag{4}$$

Values q_1, q_2, \dots ($|q_1| \leq |q_2| \leq \dots$) for which equation (4) is fulfilled are called characteristic values of the equation. The resolvent operator $R_q = [I - qA]^{-1} = A + qA^2 + \dots$ exists if the sequence converges.

Let us consider the ratio:

$$\lambda = \frac{(v, Ap(A)h)}{(v, p(A)h)} = \frac{(v, AR_q^m h)}{(v, R_q^m h)}.$$

If $q < 0$, then

$$\frac{(v, AR_q^m h)}{(v, R_q^m h)} \approx \frac{1}{q} \left(1 - \frac{1}{\mu^{(k)}} \right) \approx \lambda_n, \tag{5}$$

where $\lambda_n = \lambda_{min}$ is the minimal by modulo eigenvalue, and $\mu^{(k)}$ is the approximation to the dominant eigenvalue of R_q .

If $|q| > 0$, then

$$\frac{(v, AR_q^m h)}{(v, R_q^m h)} \approx \lambda_1, \tag{6}$$

where $\lambda_1 = \lambda_{max}$ is the dominant eigenvalue.

The approximate equations (2), (5) and (6) can be used to formulate efficient Monte Carlo algorithms for evaluating both the dominant and the minimal by modulo eigenvalue of real symmetric matrices.

2.3 Bilinear Forms of Solution of LAE Systems

Consider the bilinear form:

$$(v, x), \tag{7}$$

where x is the solution of the system:

$$Bx = b. \tag{8}$$

For a non-singular matrix B one can use the presentation of Jacobi Over-relaxation Iterative Method:

$$x = Ax + h. \tag{9}$$

Assume that matrix A satisfies:

$$\sum_{j=1}^n |a_{ij}| < 1, \quad i = 1, \dots, n. \tag{10}$$

Now, we can consider the *first-order stationary linear iterative process* for the system (9):

$$x^{(k)} = Ax^{(k-1)} + h, \quad k = 1, 2, \dots \tag{11}$$

In fact, the presentation (11) defines a Neumann series

$$x^{(k)} = h + Ah + \dots + A^{k-1}h + A^k x^{(0)}. \tag{12}$$

It is a well-known fact that the property (10) is a sufficient condition for convergence of the Neumann series, i.e.,

$$x = \lim_{k \rightarrow \infty} x^{(k)}.$$

It is clear, that every iterative algorithm (including those based on MC) uses a finite number of iterations k . If a MC algorithm is applied, then the k^{th} iteration can be computed with an additional statistical error. In practice the truncating parameter k is not a priori given parameter. Normally it is obtained from the condition that the difference between the stochastic approximation of two successive approximations is smaller than a given sufficiently small parameter ε . Thus, we approximate the bilinear form (7) by

$$(v, x) \approx \left(v, \sum_{i=0}^k A^i h \right). \tag{13}$$

One can see now, that for this problem, the matrix polynomial is of special type, i.e., $p(A) = \sum_{i=0}^k A^i$.

If the Neumann series (12) does not converge the technique of mapping can be applied. This gives us a resolvent method, with the bilinear form (7):

$$(v, x) \approx \left(v, \sum_{i=0}^k g_i^{(k)} A^i h \right). \tag{14}$$

This procedure leads to matrix polynomial of type: $p(A) = \sum_{i=1}^k g_i^{(k)} A^i$.

2.4 Matrix Inversion

Assume $B \in \mathbb{R}^{n \times n}$ is a non-singular matrix. The problem of finding the inverse matrix

$$C = B^{-1}$$

of B is equivalent to solve n times the problem (8) written in the following form:

$$Bc_j = e^{(j)}, \quad j = 1, \dots, n,$$

where $e^{(j)} \equiv (0, \dots, 0, \underbrace{1}_j, 0, \dots, 0)^T$ and $c_j \equiv (c_{1j}, \dots, c_{nj})^T$ is the j^{th} column

of the inverse matrix $C = B^{-1}$. This leads to the following bilinear form:

$$c_{ij} = (e^{(i)}, c_j) \approx \left(e^{(i)}, \sum_{i=0}^k A^i d_j e^{(j)} \right). \tag{15}$$

It is easy to see, that the latter form (15) is the same as (13) for a special choice of vectors v and h : $v = e^{(i)}$ and $h = d_j e^{(j)}$.

3 Formulation of the MC Algorithm

We shall use the so-called *MAO* algorithm, studied in [1, 2, 3, 5, 6], where

$$p_i = \frac{|v_i|}{\|v\|}, \quad \|v\| = \sum_{i=1}^n |v_i| \quad \text{and} \quad p_{ij} = \frac{|a_{ij}|}{\|a_i\|}, \quad \|a_i\| = \sum_{j=1}^n |a_{ij}|. \quad (16)$$

3.1 MC Algorithm for Computing Bilinear Forms of Matrix Powers $(v, A^k h)$

From the pair of density distributions (16) we obtain a finite chain, which induces the matrix/vector powers $A_v^k = v_{\alpha_0} \prod_{s=1}^k a_{\alpha_{s-1}\alpha_s}$ and $\|A_v^k\| = \|v\| \times \prod_{s=1}^k \|a_{\alpha_{s-1}\alpha_s}\|$. With such densities we have that $E\{h_{\alpha_k}\} = \frac{\text{sign}\{A_v^k\}}{\|A_v^k\|} (v, A^k h)$.

If we consider N realizations of the Markov chain $T_k = \alpha_0 \rightarrow \alpha_1 \rightarrow \dots \rightarrow \alpha_k$, then

$$\bar{\theta}^{(k)} = \sum_{i=1}^N \theta_i^{(k)} = \text{sign}\{A_v^k\} \|A_v^k\| \sum_{i=1}^N \{h_{\alpha_k}\}_i \quad (17)$$

is an MC approximation of the bilinear matrix power $(v, A^k h)$. The probability error of this approximation is

$$R_N^{(k)} = \left| (v, A^k h) - \bar{\theta}^{(k)} \right| = c_p \sigma\{\theta^{(k)}\} N^{-\frac{1}{2}}. \quad (18)$$

From (17), together with the sampling rules using (16), leads us to a MC algorithm for estimating $(v, A^k h)$ with a probability error $R_N^{(k)}$. The quality of the MC algorithm depends on the behaviour of the standard deviation $\sigma\{\theta^{(k)}\}$.

4 Performance Analysis

In this section we formulate the computational cost of a Monte Carlo algorithm and a deterministic algorithm for computing the bilinear forms of matrix powers $(v, A^k h)$. To do this we use the following notation: α is a cost of an addition or subtraction; β is a cost of a multiplication; and δ is a cost of a logical operation.

4.1 Computational Cost of MC Algorithm for Computing Bilinear Forms of Matrix Powers $(v, A^k h)$

To estimate the computational cost one needs to consider the following expression:

$$\text{sign}\{v_{\alpha_0}\} \sum_{\alpha} |v_{\alpha}| \left(\prod_{i=1}^k \text{sign}\{a_{\alpha_{i-1}\alpha_i}\} \sum_{\alpha_i} |a_{\alpha_{i-1}\alpha_i}| \right) h_{\alpha_k}. \quad (19)$$

The computational cost of (19) is thus:

$$\begin{aligned} & \delta + \beta + (n - 1)\alpha + \beta + (k - 1)\beta + k\delta + k(\beta + (n - 1)\alpha) + \beta \\ & = (k + 1)(n - 1)\alpha + (2k + 2)\beta + (k + 1)\delta. \end{aligned} \quad (20)$$

For the generation of the required random numbers and for the selection of the elements of v and A the following operations are required:

$$m + \delta \log n + k(m + \delta \log n), \tag{21}$$

where m is the number of operations required to generate a random number. The $\delta \log n$ term is from the binary search method [15] used to select the next element in the Markov Chain, as used in [1].

To compute the MC approximation to $(v, A^k h)$ one needs to perform N realizations of the Markov chain, so that the computational cost of the algorithm is

$$N [(k + 1)(n - 1)\alpha + 2(k + 1)\beta + (k + 1)\delta + m + \delta \log n + k(m + \delta \log n)]. \tag{22}$$

4.2 Computational Cost of a Deterministic Method for Computing Bilinear Forms of Matrix Powers $(v, A^k h)$

For a matrix-vector multiplication: $\sum_{j=1}^n a_{ij} h_j, \forall i \in 1, \dots, n$, the computational cost is:

$$n((n - 1)\alpha + n\beta) = n(n - 1)\alpha + n^2\beta.$$

For a vector-vector multiplication: $\sum_{j=1}^n v_j h_j$ the computational cost is:

$$(n - 1)\alpha + n\beta.$$

To compute the bilinear form of a matrix power, $(v, A^k h)$, one needs to perform k matrix-vector multiplications and one vector-vector multiplication. Thus, the computational cost of the deterministic method is:

$$k[n(n - 1)\alpha + n^2\beta] + (n - 1)\alpha + n\beta = (kn^2 - kn + n - 1)\alpha + (kn^2 + n)\beta. \tag{23}$$

4.3 Comparison of the Monte Carlo and Deterministic Method

For a rough estimation of the cost we may assume that $\alpha = \beta = \delta = 1$. Then, from (22), the computational cost for the Monte Carlo algorithm becomes:

$$N(kn + n + k \log n + \log n + km + m + 2k + 2) = Nkn + O(Nn) \tag{24}$$

and the computational cost for the deterministic method, from (23), is:

$$2kn^2 - kn + 2n - 1 = 2kn^2 + O(kn). \tag{25}$$

A comparison of costs (24) and (25) shows that, for a sufficiently large matrix, the Monte Carlo algorithm will be quicker than the deterministic method, provided

that one can keep the number of Markov chains required to reach a suitably accurate solution low enough. As a rough criteria for choosing the proposed Monte Carlo algorithm we can use the following inequality:

$$\frac{1}{2} \frac{N}{n} \leq 1.$$

If the required accuracy to compute $(v, A^k h)$ is ε , then (according to (18)) the following inequality

$$2\varepsilon^2 n \geq c_p^2 \sigma^2(\Theta^{(k)})$$

should be fulfilled. Let us note that for many real-life applications the matrix size n is $10^7 - 10^8$ and the typical number of Markov chains N is $10^4 - 10^5$. In such cases the Monte Carlo algorithm is definitely preferable. However, it should be mentioned that if the matrix size n is close to N , then more accurate analysis of the computational cost taking into account weights α, β and δ of different operations has to be done.

5 Conclusion

In this paper we introduced the matrix polynomial form (1). Further (2), (5) and (6) show how this form can be used to find extreme eigenvalues; (13) and (14) show how this form can be used to solve SLAE; and (15) show how to extend the solution of SLAE to MI.

From there we concentrated on the bilinear form of matrix polynomials and constructed a Monte Carlo algorithm for solving this problem. This allowed us to estimate the computational cost of the Monte Carlo algorithm, as well as the cost of the deterministic approach. These results on computational cost show that the Monte Carlo algorithm will out perform the deterministic method on sufficiently large problems, which are common to many areas of mathematical modelling. In cases when the matrix size n is close to the required number of Markov chains N a careful error analysis of the MC algorithm is needed.

References

1. V. Alexandrov, E. Atanassov, I. Dimov, *Parallel Quasi-Monte Carlo Methods for Linear Algebra Problems*, Monte Carlo Methods and Applications, Vol. 10, No. 3-4 (2004), pp. 213-219.
2. I. Dimov, *Minimization of the Probable Error for Some Monte Carlo methods*. Proc. Int. Conf. on Mathematical Modeling and Scientific Computation, Albena, Bulgaria, Sofia, Publ. House of the Bulgarian Academy of Sciences, 1991, pp. 159-170.
3. I. Dimov, *Monte Carlo Algorithms for Linear Problems*, Pliska (Studia Mathematica Bulgarica), Vol. 13 (2000), pp. 57-77.
4. I. Dimov, V. Alexandrov, A. Karaivanova, *Parallel resolvent Monte Carlo algorithms for linear algebra problems*, J. Mathematics and Computers in Simulation, Vol. 55 (2001), pp. 25-35.

5. I. Dimov, T. Dimov, T. Gurov, *A New Iterative Monte Carlo Approach for Inverse Matrix Problem*, Journal of Computational and Applied Mathematics, Vol. 92 (1997), pp. 15-35.
6. I.T. Dimov, V. Alexandrov, *A New Highly Convergent Monte Carlo Method for Matrix Computations*, Mathematics and Computers in Simulation, Vol. 47 (1998), pp. 165-181.
7. I. Dimov, A. Karaivanova, *Parallel computations of eigenvalues based on a Monte Carlo approach*, Journal of Monte Carlo Method and Applications, Vol. 4, Nu. 1, (1998), pp. 33-52.
8. I. Dimov, A. Karaivanova, *A Power Method with Monte Carlo Iterations*, Recent Advances in Numerical Methods and Applications, (O. Iliev, M. Kaschiev, Bl. Sendov, P. Vassilevski, Eds.), World Scientific, 1999, Singapore), pp. 239-247.
9. G.V. Golub, C.F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins Univ. Press, Baltimore, 1996.
10. I.M. Sobol *Monte Carlo numerical methods*, Nauka, Moscow, 1973.
11. J.R. Westlake, *A Handbook of Numerical matrix Inversion and Solution of Linear Equations*, John Wiley & Sons, inc., New York, London, Sydney, 1968.
12. I. Dimov, O.Tonev, *Monte Carlo Algorithms: Performance Analysis for Some Computer Architectures*, Journal of Computational and Applied Mathematics, Vol. 48 (1993), pp. 253-277.
13. I. Dimov, *Monte Carlo Algorithms for Linear Problems*, Pliska (Studia Mathematica Bulgarica), Vol. 13 (2000), Proceedings of the 9th International Summer School on Probability Theory and Mathematical Statistics, Sozopol, 1997, pp. 57-77.
14. I. Dimov, A. Karaivanova and P. Yordanova, *Monte Carlo Algorithms for calculating eigenvalues*, Second International Conference on Monte Carlo and Quasi-Monte Carlo methods in scientific computing, University of Salzburg, 8-12 July, 1996, (in: Proceedings of MC & QMC 96, Springer Notes in Statistics (H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, Eds)), 1997, pp. 205-220.
15. D. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, Third Edition. Addison-Wesley, 1997.

Performance Analysis of the Cache Conscious-Generalized Search Tree

Won-Sik Kim¹, Woong-Kee Loh², and Wook-Shin Han^{1,*}

¹ Department of Computer Engineering
Kyungpook National University, Korea
wskim@www-db.knu.ac.kr, wshan@knu.ac.kr

² Department of Computer Science &
Advanced Information Technology Research Center (AITrc)
Korea Advanced Institute of Science and Technology (KAIST), Korea
woong@mozart.kaist.ac.kr

Abstract. Recently, a main memory index structure called the cache conscious-generalized search tree (CC-GiST) was proposed. The CC-GiST is such a novel index structure that it can be used for implementing all the existing cache conscious trees with the minimal efforts. It incorporates the pointer compression and the key compression techniques, which were adopted by the existing cache conscious trees to reduce the cache misses, in a single framework. In this paper, we formally analyze the performance of the CC-GiST. We compare the performance of the CC-GiST with the existing cache conscious trees. The result shows that the CC-GiST has the negligible overhead for supporting all the existing cache conscious trees in a single framework, and the performance of the tree is almost unaffected.

Keywords: cache conscious tree, generalized search tree, pointer compression, key compression.

1 Introduction

According to the advance of technologies, the speed gap between CPU and main memory is getting larger every year [5, 10]. Due to the speed gap, it was perceived important to make the most use of the cache residing between CPU and main memory, and there have been a lot of research efforts on this issue [1, 3, 4, 5, 6]. Among those is the research on cache conscious trees for reducing the cost for accessing main memory indexes [2, 8, 10]. Cache conscious trees were designed to cause as few cache misses as possible based on the characteristics of the cache. The most widely known cache conscious trees are the CSB⁺-tree [10], the pkB-tree [2], and the CR-tree [8].

Recently, we proposed the cache conscious-generalized search tree (CC-GiST) [9], which is an extension of the disk based GiST [7]. The CC-GiST can be used for implementing all the existing cache conscious trees with the minimal efforts.

* Corresponding author.

It incorporates the pointer compression and the key compression techniques, which were adopted by the existing cache conscious trees to reduce the cache misses, in a single framework. In this paper, we provide the formal analysis on the performance of the CC-GiST. We compare the performance of the CC-GiST with the existing cache conscious trees. As the result, the CC-GiST has the negligible overhead for supporting all the existing cache conscious trees in a single framework, and the performance of the tree is almost unaffected.

The rest of this paper consists of the following. In Section 2, we briefly explain on the existing cache conscious trees and the CC-GiST. In Section 3, we analyze the performance of the CC-GiST. Finally, we conclude this paper in Section 4.

2 Related Work

In this section, we analyze the existing cache conscious trees and derive the pointer compression and the key compression techniques used by the trees. We also briefly explain the CC-GiST [9].

2.1 Cache Conscious Trees

The cache conscious trees use the techniques for reducing the memory size of the indexes and for increasing the blocking factors of the nodes in the indexes. We call the techniques as *compressions* in this paper. In general, when the blocking factor gets larger, since it is more probable that the related data are contained in the same cacheline, it can reduce the cache misses¹. We can group the existing cache conscious trees into two categories according to their compression techniques: (1) those based on pointer compression and (2) those based on key compression. The cache conscious trees based on pointer compression remove the subset of pointers to child nodes and thus increase the blocking factors in the internal nodes. The examples are the Cache Sensitive B⁺-tree (CSB⁺-tree) [10] and the Segmented CSB⁺-tree [10]. The cache conscious trees based on key compression store the keys of shorter sizes and thus increase the blocking factors in the internal and the leaf nodes. The examples are the Partial Key B-tree (pkB-tree) [2] and the Cache Conscious R-tree (CR-tree) [8].

Figure 1 shows an example of the CSB⁺-tree. All the child nodes of an internal node are stored in a *node group*, which is represented by a dashed rectangle in the figure, and the internal node manages only one pointer to the first child node in the node group, which is represented by a solid arrow in the figure. Since the child nodes in a node group are stored physically consecutively in main memory, the pointer to any child node in the node group can be easily found even by managing only one pointer. In this manner, the performance of the tree can be enhanced by reducing the number of pointers and thus increasing the blocking factor.

¹ The data in main memory are loaded into cache in the unit of cacheline. The usual size a cacheline is 64 ~ 128 bytes, and the size of a node in cache conscious trees is the integer multiple of the size of cacheline.

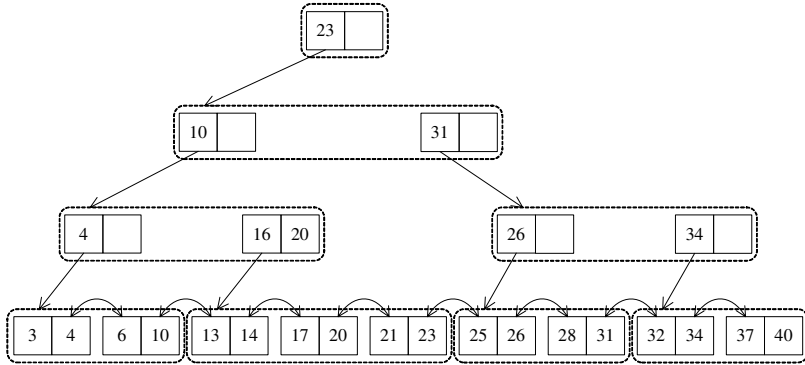


Fig. 1. An Example of the CSB⁺-Tree

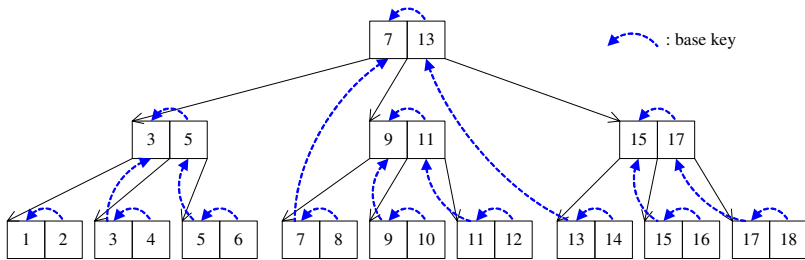


Fig. 2. An Example of the pkB-Tree

The CSB⁺-tree has a problem of increased update cost. When there happens a split or a merge of nodes while inserting or deleting data in the tree, to maintain the child nodes in a node group stored physically consecutively even after the split or the merge, all the nodes stored in the node group should be copied. The Segmented CSB⁺-tree is a variant of the CSB⁺-tree to tackle the problem, and stores the child nodes of an internal node in one or more *segments* rather than only one node group to reduce the copy cost. Only the child nodes in the same segment are stored physically consecutively, and the internal node manages only the pointers to the first nodes of the segments. The CSB⁺-tree in Figure 1 can be regarded as a Segmented CSB⁺-tree that has only one segment in each internal node.

The pkB-tree, which is based on key compression, manages only the different part between adjacent keys in a node [2]. The tree stores only the short part of keys rather than the long whole keys, which results in the increasing of the blocking factor of the node. Figure 2 shows an example of the pkB-tree. The solid arrow represents the pointer to a child node, and the dashed arrow represents the *base key* that is used for key compression. The pkB-tree performs key compression by managing only the different part between the base key and the key to be compressed.

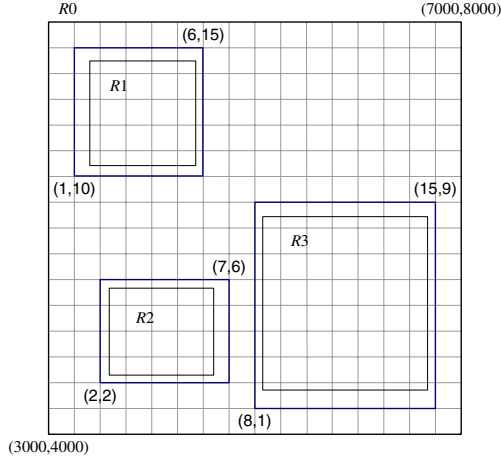


Fig. 3. An Example of QRMBRs in the CR-Tree

The CR-tree, which is also based on key compression, manages the Quantized Relative Representation of MBR (QRMBR) rather than the actual MBR [8]. The QRMBR is generated by quantizing the relative coordinates of the actual MBR to the parent node's MBR. Since the coordinates of the QRMBR have smaller size than those of the actual MBR, the CR-tree stores more entries in a node. Figure 3 shows an example of QRMBRs in the CR-tree. The outermost rectangle R_0 is the parent node's MBR, and the inner rectangles R_1 , R_2 , and R_3 are child nodes' MBRs. The thick rectangles surrounding the child nodes' MBRs are their QRMBRs generated relatively to R_0 . Since the relative coordinates are quantized in 16 units, which require only four bits, the space to store the coordinates can be reduced.

2.2 Cache Conscious-Generalized Search Tree (CC-GiST)

The CC-GiST is a balanced search tree with the blocking factor between kM and M , where k is the minimum fill factor ($\frac{2}{M} \leq k \leq \frac{1}{2}$), and M is the maximum blocking factor in the tree. An exception is the root node that has the blocking factor between 2 and M .

Based on the pointer compression technique, child nodes are stored in segments, and parent nodes manage only the pointers to the segments in the CC-GiST. A segment contains one or more adjacent child nodes stored physically consecutively in main memory. The CC-GiST supports both the fixed- and the variable-size segments. Whether to use the fixed- or the variable-size segment is determined by the flag `IsFixedSizeSegment`. The CC-GiST manages the variable `MaxNumOfNodesInSegment` representing the maximum number of child nodes contained in a segment.

An internal node in the CC-GiST manages `ListOfSegmentPointers`, a list of pointers to the segments of the node. It also manages `ListOfNumOfChildNodesInSegment`, a list of the numbers of the child nodes in each segment of the node.

A pointer to a child node of an internal node in the CC-GiST, which is represented as $\text{NodePointer} = (\text{SegmentPointer}, \text{NodeOrderInSegment})$, consists of a pointer to the segment containing the child node and the order of the child node in the segment ($\text{NodeOrderInSegment} \geq 0$). The NodePointer is used to specify a child node of an internal node, and is not actually stored in the CC-GiST.

Based on the key compression technique, the CC-GiST manages the flag $\text{StoreBaseKeyInNode}$ to indicate whether to contain the space to store the base key in a node as in the CR-tree. If the flag is set to TRUE , every node in the CC-GiST contains the space for the base key. The CC-GiST provides an object AncestorKeyStack to store the base keys from the ancestor nodes on the search path used when compressing and decompressing the keys. The AncestorKeyStack is temporarily used while searching, insertion, and deletion, and is not actually stored in the CC-GiST. A key in a node in the CC-GiST is represented by a predicate as in the GiST. A key predicate has one or more free variables and becomes TRUE when the variables are instantiated with the values of any tuple reachable from the pointer associated with the predicate. A node in the CC-GiST manages a list of key predicates.

3 Performance Analysis

In this section, we analyze the performance of the CC-GiST [9]. Given the node size, the performance of the cache conscious trees is inversely proportional to the average number of entries in a node. The less the number of entries is, the more probably the cache misses are incurred, which causes the increase of main memory accesses and thus the performance degradation of the cache conscious trees. Therefore, the existing cache conscious trees tried to increase the number of entries in a node using the pointer and the key compression techniques. In this section, we compare the number of entries in an internal node in the cache conscious trees implemented using the CC-GiST with those in the existing cache conscious trees.

Figure 4 shows the structure of an internal node in the CC-GiST. The internal node consists of a header, a base key, a key list, a list of the numbers of child nodes in segments, and a list of segment pointers. The header contains the number of keys (M), the number of segments (N), and a few flags. The base key can be excluded in the node according to the cache conscious tree implemented using the CC-GiST. In the figure, the dashed arrows indicate the directions of inserting new keys and segments.

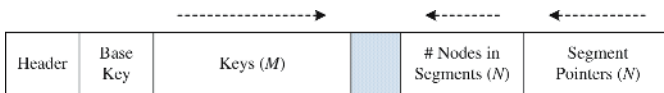


Fig. 4. Structure of an Internal Node in the CC-GiST

The number of entries in an internal node in the CC-GiST should be computed according to whether to perform the pointer compression or not. When the pointer compression is performed ($M > N$), the number of entries is as the following:

$$\frac{NodeSize - (HeaderSize + BaseKeySize + SegNodesListSize + SegPtrListSize)}{KeySize}, \quad (1)$$

where $SegNodesListSize$ is the size of the list of numbers of child nodes in segments, and $SegPtrListSize$ is the size of the list of segment pointers. Without pointer compression, since each of the keys is associated with a segment pointer ($M = N$), the number of entries is as the following:

$$\frac{NodeSize - (HeaderSize + BaseKeySize)}{KeySize + SegNodesSize + SegPtrSize}, \quad (2)$$

where $SegNodesSize$ is the size for storing the number of child nodes in a segment, and $SegPtrSize$ is the size of a segment pointer.

The Lemmas 1, 2, and 3 show the ratios computed by dividing the numbers of entries in the internal nodes in the cache conscious trees implemented using the CC-GiST by those in the existing cache conscious trees: each for the CSB⁺-tree [10], the pkB-tree [2], and the CR-tree [8]. In the Lemmas, it is assumed that the header size $HeaderSize$ is four bytes, the size of a segment pointer $SegPtrSize$ is four bytes, and the size for storing the number of child nodes in a segment $SegNodesSize$ is one byte. For the proofs for the Lemmas, please refer to [9].

Lemma 1. For any key size, the average ratio computed by dividing the number of entries in an internal node in the existing CSB⁺-tree by that in the CSB⁺-tree implemented using the CC-GiST is as follows:

$$1 + \frac{3}{NodeSize - 9}. \quad (3)$$

□

Lemma 2. The average ratio computed by dividing the number of entries in an internal node in the existing pkB-tree by that in the pkB-tree implemented using the CC-GiST is as follows:

$$\left(1 - \frac{2}{NodeSize - 4}\right) \cdot \frac{14}{13}. \quad (4)$$

□

Lemma 3. The average ratio computed by dividing the number of entries in an internal node in the existing CR-tree by that in the CR-tree implemented using the CC-GiST is as follows:

$$\left(1 + \frac{2}{NodeSize - 20}\right) \cdot \frac{7}{6}. \quad (5)$$

□

Figure 5 shows the ratios of the numbers of entries for each of the cache conscious trees given in Lemmas 1, 2, and 3 as we vary the node size. The node sizes in the figure are determined as the multiples of minimal cacheline size (64 bytes). As shown in the figure, the numbers of entries in the internal nodes in the cache conscious trees implemented using the CC-GiST are not increased much compared with those in the existing cache conscious trees. Especially, for the CSB⁺-tree and the pkB-tree implemented using the CC-GiST, the ratios are increased no more than 7% for all the node sizes. The CR-tree implemented using the CC-GiST has the relatively larger ratios than the other trees. However, if we don't manage the number of child nodes in a segment, whose size is *SegNodesSize* in Eq. (2), the ratio will be dropped.

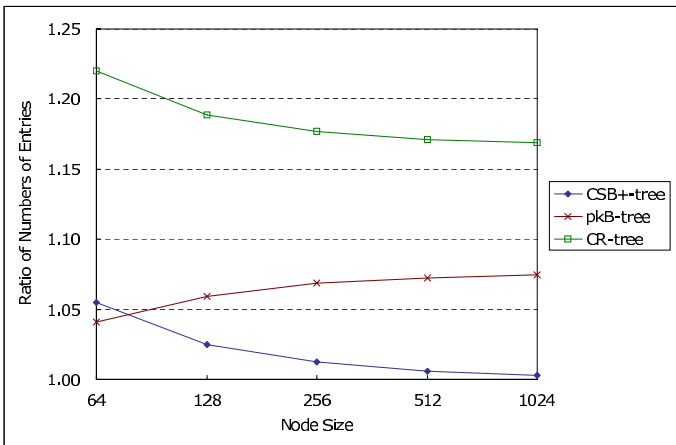


Fig. 5. Ratios of Numbers of Entries

As the result, the CC-GiST has the negligible overhead for supporting all the existing cache conscious trees in a single framework, and the performance of the tree is almost unaffected. We analyzed the ratios of the numbers of entries only for the internal nodes in this section; however, since the leaf nodes of the CC-GiST have almost the same structures as those of the existing cache conscious trees, we expect to have almost the same performance for the leaf nodes.

4 Conclusions

In this paper, we provided the formal analysis on the performance of the CC-GiST. As the result, the CC-GiST has the negligible overhead for supporting all the existing cache conscious trees in a single framework, and the performance of the tree is almost unaffected. The CC-GiST can be used not only to easily implement the existing cache conscious trees but also to develop a new tree integrating all the advantages of the existing cache conscious trees.

Acknowledgement

This work was supported by Korea Research Foundation Grant (KRF-2003-003-D00347).

References

1. A. Ailamaki, D. J. DeWitt, M. D. Hill, and D. A. Wood, "DBMS on a Modern Processor: Where Does Time Go?," In *Proc. Int'l Conf. Very Large Databases*, pp. 54-65, Edinburgh, Scotland, UK, Sept. 1999.
2. P. Bohannon, P. Mcilroy, and R. Rastogi, "Main-Memory Index Structures with Fixed-Size Partial Keys," In *Proc. ACM SIGMOD/PODS Int'l Conf. Management of Data*, pp. 163-174, Santa Barbara, California, May 2001.
3. P. A. Boncz et al., "Database Architecture Optimized for the New Bottleneck: Memory Access," In *Proc. Int'l Conf. Very Large Databases*, pp. 54-65, Edinburgh, Scotland, UK, Sept. 1999.
4. B. Calder, C. Krintz, S. John, and T. Austin, "Cache-Conscious Data Placement," In *Proc. 8th Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 139-149, Oct. 1998.
5. T. M. Chilimbi, J. R. Larus, and M. D. Hill, Improving Pointer Based Codes through Cache-Conscious Data Placement, Technical Report, Computer Science Department, University of Wisconsin-Madison, 1998.
6. T. M. Chilimbi, J. R. Larus, and M. D. Hill, "Making Pointer Based Data Structures Cache Conscious," *IEEE Computer*, Vol. 33, No. 12, pp. 67-74, Dec. 2000.
7. J. M. Hellerstein, J. F. Naughton, and A. Pfeffer, "Generalized Search Trees for Database Systems," In *Proc. Int'l Conf. Very Large Data Bases*, pp. 562-573, Zurich, Switzerland, Sept. 1995.
8. K. Kim, S. K. Cha, and K. Kwon, "Optimizing Multidimensional Index Trees for Main Memory Access," In *Proc. ACM SIGMOD/PODS Int'l Conf. Management of Data*, pp. 139-150, Santa Barbara, California, May 2001.
9. W.-S. Kim, W.-K. Loh, and W.-S. Han, CC-GiST: Cache Conscious-Generalized Search Tree, Technical Report, Department of Computer Engineering, Kyungpook National University, 2006 (Also available at <http://www-db.knu.ac.kr/~wshan/cc-gist.pdf>).
10. J. Rao and K. A. Ross, "Making B⁺-Trees Cache Conscious in Main Memory," In *Proc. ACM-SIGMOD Int'l Conf. Management of Data*, pp. 475-486, Dallas, Texas, May 2000.

A Database Redo Log System Based on Virtual Memory Disk*

Haiping Wu, Hongliang Yu, Bigang Li, Xue Wei, and Weimin Zheng

Department of Computer Science and Technology, Tsinghua University,
100084, Beijing, P.R. China
{wuhp, hlyu, lbg01, xuwei, zwm-dcs}@tsinghua.edu.cn

Abstract. Redo log of database must be written to permanence storage like disks. When database is heavily loaded, the crowded redo log writing queue will become a performance bottleneck. In this paper, the operation principle of database redo log is analyzed. It is found that if redo log is stored on virtual memory disk directly, the database will demonstrate better performance. In addition, the reliability of redo log system based on virtual memory disk is analyzed. At the end of the paper, a contrastive performance measurement result is given.

1 Introduction

Modern database is capable of storing several TB (10^{12}) data. But it is still necessary to improve a database's performance while gaining a much larger capacity. Traditional redo log of database is stored in permanence storage like hard disks. When redo log is created, it is written in redo log buffer firstly. After redo log buffer is filled up, items of redo log are moved from buffer to log files stored in disk. Because the I/O speed of disk is much slower than CPU and memory, frequent writing operation of redo log files will become the performance bottleneck [1].

Many large databases are running on storage network area (SAN). General speaking, SAN has high performance in cluster systems. On the other hand, the redundant memory of clusters' node servers can be separated to be a virtual memory disk. The capacity of virtual memory disk can only reach 4GB in IA-32 servers; and using an IA-64 server, 100GB capacity can be gotten. It is for sure that the capacity of virtual memory disk is not enough to store all massive data of database. But, the small redo log files which are frequently written can be stored in virtual memory disk. It can improve a database's performance.

This paper describes operating principle of database's redo log in section 2. Then, a virtual memory disk based on SAN is designed. Because memory is more vulnerable than traditional disks, mirror technique is used. The reliability analysis indicates it can be used to store redo log files. At last, two databases' performances are compared, with redo log files of one database stored in virtual memory disk and

* This research was partially supported by the Chinese "863" project, sponsored by the Chinese Ministry of Science and Technology under contract 2003AA1Z2330.

those of the other are stored in SAN's permanence disk. The result indicates that the performance of the former is better than the latter.

2 Summary of Database's Redo Log

Information about database's transaction is recorded by the redo log. Each item of redo log describes database users' writing operation: WRITE(Q). The redo log record is usually composed of name of transaction, name of data item Q, original value of Q before WRITE(Q) is commit and new value of Q after WRITE(Q) is commit.

Redo log also includes other items such as beginning point of transaction, end point of transaction, etc. Before a transaction commits WRITE(Q), the redo log items are created. When needed, the value of Q stored in redo log can update the value of Q stored in database, or, can be used to recover the original value [2].

Because redo log should be always usable, it must be stored in permanence storage equipment such as hard disks. For example, each Oracle database has two or more online redo log files. Oracle writes these files by circulation way: after the first file is fulfilled, redo log items are written in the second file, then, the rest may be deduced by analogy. When all online redo log files are used up, the first redo log file is written again. The database can be recovered to any point by using redo log files [3].

Redo log is so important that it must be written in permanence disk before the transaction is committed. As the size of a redo log item is much smaller than I/O unit of disk, writing single redo log item is expensive. So, the redo log buffer is created in memory and the items of redo log are written in buffer firstly. After buffer is filled up, items of redo log are moved from buffer to log files stored in disk. More over, the disk is a sort of mechanism. Its speed is far slower than CPU or memory. Frequent writing operation of redo log files will become the performance bottleneck. Acceleration of writing redo log becomes more important [4].

3 Virtual Memory Disk Based on SAN

3.1 Design of Virtual Memory Disk

With the development of data storage technique, many new storage structures come into being. SAN is a new representative storage structure. It is composed of storage control cluster, massive disk pool and high speed network. Application server which connects SAN can access data stored in network disk pool through high speed network. Compared with traditional disk, SAN is more flexible and more reliable. It can provide higher performance storage with much larger capacity.

Shu[5] has designed a SAN system. In order to improve the performance and reliability, storage control server usually adopts a cluster which is composed of several nodes. Every node in the cluster owns a big memory. The redundant memory of clusters' nodes can be separated to form a virtual memory disk. Figure 1 shows the structure of virtual memory disk.

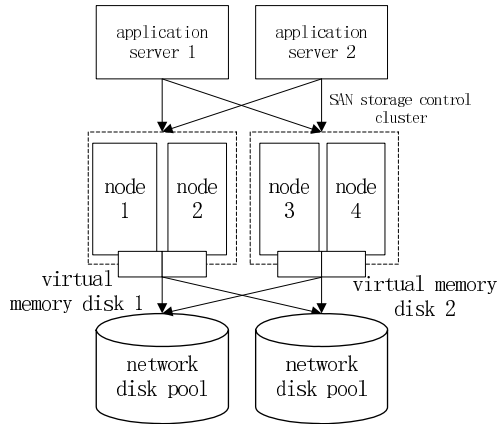


Fig. 1. Structure of virtual memory disk based on SAN

The interface between virtual memory disk and server is the same as general hard disk. The data is read from or written in memory disk directly. Because virtual memory disk is embedded in server’s memory, the time which is spent finding track of disk is saved.

3.2 Reliability Analysis

Memory is more vulnerable compared with the traditional disk. When server’s power is off or there is a technical problem in writing to files on the storage medium, data in the virtual memory disk will be lost. Any node server in the storage control cluster of SAN may encounter failure at any time, including hardware failure, software failure and user error. If it can’t recover on time after these failures appear, the usability of system would decline greatly.

An easy solution is making a hard copy in permanence disk while data is written in virtual memory disk. Its reliability is not very good, for data’s writing is asynchronous and data’s consistency can not be kept well. This method will also affect data’s writing performance.

The better way is using mirror disk and snapshot disk technique to improve virtual memory disk’s reliability. In this method, the node servers in the storage control of SAN are divided into N groups, with every group composed of two nodes. The virtual memory disk of these two nodes are mirrored each other. When an error occurs in one node, the other node will provide virtual memory disk service in place of it and at the same time, one hard copy is made in performance disk of SAN. If two nodes in the same group encounter failure concurrently, data will be read from or written in the hard copy through other groups in the cluster. At this time, the performance of virtual memory disk will be the same as that of permanence disk. When all nodes in the cluster fail, data will be lost in virtual memory disk. Figure 2 shows mirror and snapshot of virtual memory disk.

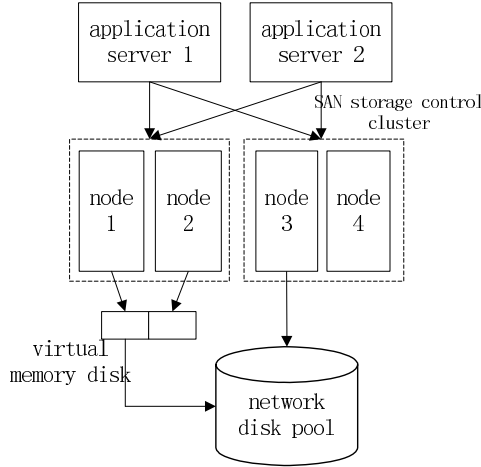


Fig. 2. Mirror and snapshot copy of virtual memory disk

It is supposed that average failure cycle of every node in the cluster is 1 week, and it takes 1 minute to recover. When the cluster has 8 nodes which can be divided into 4 groups, the chance of 8 nodes' concurrent failure is 10^{-30} , which can be ignored.

4 Database Redo Log System Based on Virtual Memory Disk

4.1 Design of Redo Log File System

Figure 3 shows a database redo log system based on virtual memory disk. The SAN in the figure 3 has four node servers and is divided into two groups, with each group

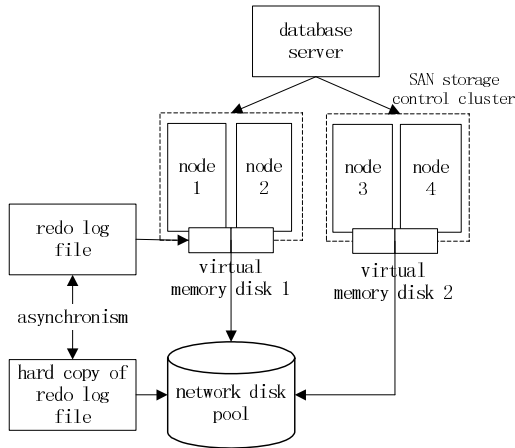


Fig. 3. Database's redo log system based on virtual memory disk

having two nodes. Some memory of node server has been partitioned and forms virtual memory disk. The virtual memory disk of two nodes in the same group are mirrored each other. So, there are two virtual memory disks in figure 3. The database redo log files are stored in these virtual memory disks.

Although virtual memory disk has better performance than permanence disk, it is more vulnerable. In order to prevent its data from being lost when servers' power is off, a hard copy in permanence network disk pool is created. When node server is idle, it writes redo log file in the virtual memory disk to its hard copy in network disk pool of SAN.

There is a log buffer in a traditional database. When a virtual memory disk is used, the disk's performance is the same as that of memory. So, redo log buffer can be canceled.

4.2 Performance Analysis

The performance of a hard disk is greatly influenced by delay time greatly. Delay time includes tracking time and rotating time. A disk rotates to find data's sector in rotating time and find data's track in tracking time. Formula 1 shows a disk's delay time [6]:

$$T_{write_disk_rand} = aL_{data} + b \tag{1}$$

Coefficient A is a constant. When an accessing disk's pattern is fixed, coefficient B can be taken as a constant, too.

Delay time of virtual memory disk mainly includes network transportation time, memory writing time and node processing time. It is expressed in formula 2:

$$T_{write_mem_band} = T_{data} + T_{mem-write} + T_{operation} \tag{2}$$

In formula 2, $T_{operation}$ is node processing time; $T_{mem-write}$ is memory writing time. They are decided by performance of a cluster's node servers and can be taken as a constant. T_{data} is network transportation time, and is proportional to size of data package.

For testing the performance of redo log system stored in virtual memory disk, an Oracle database is installed in TH_MSNS[5], a self-designed SAN system. Performances are compared when redo log file is stored in virtual memory disk and permanence disk. The system configurations can be seen from Table 1.

Table 1. System configurations

Storage Capacity	4T
Network bandwidth in the SAN	2Gbps
Operating system	Redhat Linux 2.4.18
Database	Oracle9i
Database server	Intel Pentium4 Xeon 2.4Ghz, 2Gb memory
Storage control cluster node	Intel Pentium4 Xeon 2.4Ghz, 2Gb memory

From formulae 1 and 2, it is known that delay time is related to the size of data package. Database data block can be 4KB, 8KB, 16KB, and 32KB, etc. According to

the size of data block, delay time is tested separately. The result is lineal approximation, which can be expressed in the following formulae:

$$T_{write_disk_rand} = 0.27L_{data} + 66 \quad (3)$$

$$T_{write_disk_rand} = 0.064L_{data} + 0.025 \quad (4)$$

From formula 3 and 4, the performance of a virtual memory disk is much better than a permanence disk. When data package size is 4KB, the ratio of delay time of a permanence disk and a virtual memory disk is 1:237; when data package size is 64KB, the ratio is 1:20. It is obvious that the smaller data package size is, the better performance of a virtual memory disk is.

The database's performance is measured by Benchmark Factory V3.3 which is developed by Quest Corporation. The tool adopts TPC-C benchmark established by Transaction Processing Performance Council. As an OLTP system benchmark, TPC-C simulates a complete environment where a population of terminal operators executes transactions against a database. The benchmark is centered on the principal activities (transactions) of an order-entry environment. These transactions include entering and delivering orders, recording payments, checking the status of orders, and monitoring the level of stock at the warehouses. We simulate 100, 200, 300, 400 and 500 users to access database concurrently and get database's transaction numbers in time unit. The result is shown in Table 2 and 3.

Table 2. The transaction number in time unit when stored in permanence disk

Users	4KB	8KB	16KB	32KB	64KB
100	380	434	450	471	506
200	762	870	896	1206	1208
300	1151	1291	1390	1415	1524
400	1528	1598	1673	1737	1895
500	1873	2187	2184	2289	2409

Table 3. The transaction number in time unit when stored in virtual memory disk

Users	4KB	8KB	16KB	32KB	64KB
100	426	473	485	501	528
200	847	942	1062	1209	1284
300	1275	1402	1488	1537	1603
400	1701	1823	1836	1906	2018
500	2104	2309	2298	2413	2536

It is obvious that the transaction number increases with increase of the number of users. The performance of the system improves lineally. The result also indicates that the performance is better when redo log files are stored in a virtual memory disk rather than in a permanence disk. When data block is 4KB, the average ratio of

improvement is 1.12: 1; when data block is 8KB or 16KB, the average ratio is 1.09: 1; when data block is 32KB or 64KB, the average ratio is 1.06: 1.

Because the database performance is not only related to redo log files, but also related to the size of data block buffer. Data block buffer is used to store data which database users usually access to. So, the bigger a data block is, the more beneficial it is to improve a database's performance. We can see from the experiment that when the block size is increased, the ratio of the performance gained from virtual memory disk declines.

5 Conclusion

When a database is heavily loaded, frequent writing operation of redo log files will become the performance bottleneck. This paper designs a database redo log file system which is stored in a virtual memory disk. The average ratio of a database's performance get an improvement of 6 to 11 percent when redo log files are stored in a virtual memory disk compared with a permanence disk.

In traditional database, the system tablespace and rollback segment are also the factor which influences database's performance. It is also possible to store them in a virtual memory disk, and more performance gains can be expected. Further studies need to be carried out in terms of this.

References

1. Lomet, David B.: Persistent applications using generalized redo recovery. Proceedings of the 1998 14th International Conference on Data Engineering, (1998) 154-163.
2. Haerder, Theo, Reuter, Andreas.: Optimization of logging and recovery in a database system. ACS Symposium Series, (1979) 151-168
3. Lance Ashdown, Valarie Moore.: Oracle9i Backup and Recovery Concepts, Release 2 (9.2). Oracle Corporation. Oracle Parkway, Redwood City (2002).
4. Kumar, Vijay, Moe, Shawn, D.: Performance of recovery algorithms for centralized database management systems. Information Sciences, V.86, n1-3, (1995) 101-147.
5. Shu, J.W., Xue, W., Li, B.G., Zheng, W.M.: TH-MSNS: A High Scalable Network Storage System. Chinese Journal of Computers. V.28, n3, (2005), 326-333.
6. Jin C., Zheng, W.M., Mao, Y., Wang, D.S.: Two methods to improve the performance of synchronous disk I/O in Linux environment. The Third LCI International Conference on Linux Clusters, St. Petersburg, USA, 2002.

Design and Implementation of an Out-of-Band Virtualization System on Solaris 10*

Yang Wang¹, Wei Xue², Ji-Wu Shu³, and Guang-Yan Zhang⁴

Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China

iodine01@mails.tsinghua.edu.cn,
xuewei@mail.tsinghua.edu.cn,
shujw@tsinghua.edu.cn,
zhang-gy04@mails.tsinghua.edu.cn

Abstract. In out-of-band virtualization systems, it is typical that the virtual storage manager (VSM) maintains the metadata for all Agents and performs the address mapping. The Agent performs I/O access according to the mapped address. High network latency, poor performance and low scalability are the main problems of this method. This article introduces an improved design and implementation of an out-of-band virtualization system, in which each Agent maintains a copy of the metadata and performs the address mapping by itself, so as to improve performance, reliability and scalability. This article discusses the on-line extension of a logical volume and the related problems such as synchronization. This function improves the ability of uninterrupted service. The system was tested on the UFS of Solaris 10 and the results showed that the performance of a stripe volume consisting of four disks exceeded that of a linear volume by an average of 103.65%.

1 Introduction

Storage virtualization is an important technique in Storage Area Networks (SANs). It brings the following benefits: 1) Reducing machine downtime [1]. 2) Increasing storage resource utilization[1]. The disk utilization can be increased up to 80% through the administration of virtualization software [2]. 3) A virtual volume can become much larger than the size of a single disk or even than a single RAID-system [2]. 4) Offering new degrees of flexibility. Storage systems can be added to or removed from storage pools without downtime [2]. We designed and implemented a virtualization solution for better utilization of our SAN system, TH-MSNS[3].

Storage virtualization system can work in two ways: in-band and out-of-band [4]. In an in-band virtualization system, data and control information flow in the same path. Data transmission and address mapping are both done by the Virtual Storage

* Supported by the National Natural Science Foundation of China under Grant Nos.60473101 and 10576018; and the National Grand Fundamental Research 973 Program of China under Grant No. 2004CB318205.

Manager (VSM). In an out-of-band virtualization system, the VSM is located outside of the data path, so as to improve performance and scalability, but different Agents are required on different platforms [5,6]. HP's OpenView [7] is a typical out-of-band virtualization system. It implements an Agent in the HBA driver on the host. OpenView is limited to a given HBA card and driver, and has poor compatibility. Our system implements an Agent on the volume manager level, universal for different kinds of HBA cards and drivers.

In a typical out-of-band virtualization system, the VSM maintains metadata and uses it to perform address mapping. The Agent has to query the VSM in each I/O access for address mapping. Our experiment has shown that this method brings too much communication latency, thus reducing the throughput of the disks. Therefore, we introduced an improved design of an out-of-band virtualization system [2]: the Agent gains a copy of the metadata from the VSM and conducts address mapping by itself. This method provides the following benefits: 1) High performance: the Agent communicates with the VSM only when getting the copy of metadata and does not require any communication in I/O accesses. The virtual volume can have a performance level comparable with the local disk. 2) High reliability: the Agent maintains a copy of the metadata. Even if the VSM fails for some reason, the Agent can still work. 3) High scalability: the work of address mapping is distributed to every Agent. The VSM has low overhead and is able to manage more Agents and logical volumes.

Volume Extension, which means enlarging a logical volume without damaging its data, is an important characteristic of virtualization systems. Volume Extension includes not only the updating of metadata, but also the refreshment of the file system. Therefore, Volume Extension requires support from both virtualization software and OS and its file system. Some third party software such as PowerQuest Partition Magic can change the size of a volume without damaging its data, but they normally work off-line, requiring rebooting or terminating of other applications. A modern storage system is required for 7×24 on-line time [8], so off-line extension cannot satisfy the requirement. Solaris provides a lot of support for on-line extension, but it is not integrated in the virtualization system and needs complex management. This article researches how to extend a logical volume automatically with support from both the virtualization and operating system. We implemented an on-line extension function on Solaris 10 and UFS (Unix File System).

This article introduces the design and implementation of the virtualization system on Solaris 10. Section 2 introduces the design of VSM. Section 3 describes the design and implementation of the Agent on Solaris 10. Section 4 presents the design and implementation of on-line extension. Finally, the test results of the linear and stripe volumes on UFS are shown.

2 Design of VSM

The VSM is responsible for the management of metadata. The design of the VSM is shown in Fig.1, consisting of five modules:

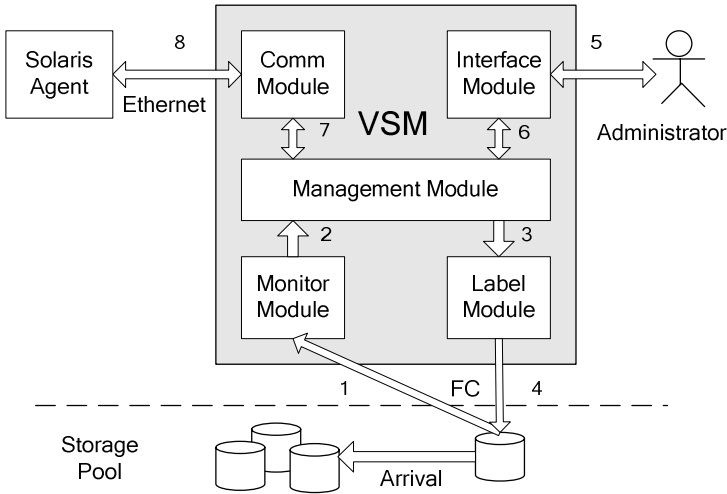


Fig. 1. Design of the VSM

The Management Module is the core of the whole system, maintaining metadata for all Agents and controlling and coordinating other modules.

The Monitor Module watches the status of all storage devices. When a storage device is added or removed, the Monitor Module will inform the Management Module (process 1 and 2).

The Label Module writes an identifier (UUID) at a specific location of the physical disk. It is used to identify the same physical disk among different machines. When the Monitor Module detects the arrival of a new disk, the Management Module will inform the Label Module to do this process (process 3 and 4). This UUID is used as a global name for this physical disk in messages between the VSM and the Agent.

The Interface Module processes the administrator's commands and displays the VSM's information (process 5 and 6). Two methods are provided: command line and GUI, and its main tasks are command parsing and information displaying.

The Comm Module is responsible for the communication between the VSM and Agent. The main content of communication is the metadata and the returning information after executing commands (process 7 and 8). Communication is based on Ethernet and TCP.

3 Design and Implementation of Agent on Solaris

The Solaris Agent consists of a Comm Module in user space and a Driver Module in kernel space as shown in Fig.2. The Comm Module receives commands from the VSM, translates them and sends them to the Driver Module. The Driver Module performs the address mapping and consists of three parts: the metadata of a logical volume—LV, the accessing point for the Comm Module—admin pseudo device and the address mapping and dispatching module—I/O.

Implementing the Comm Module in user space can improve the reliability and portability of the whole system, but it brings extra inter-process communication overhead. The Comm Module works only when updating metadata, and the frequency of metadata updates is much lower than the frequency of disk accesses, so this overhead will not affect the I/O performance of a logical volume.

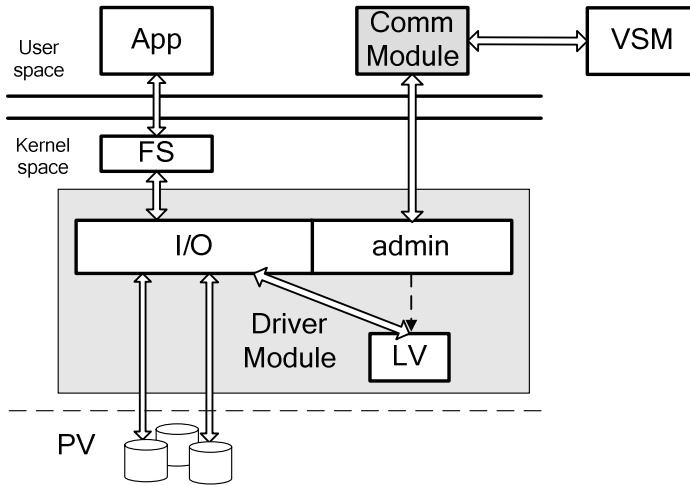


Fig. 2. Design of the Agent

3.1 Comm Module

The Comm Module receives commands from the VSM, checks their validity, and then translates them into formats that can be understood by Driver Module. It sends them to the Driver Module through the admin pseudo device and finally returns the executing result to the VSM. The commonly used commands include creating a logical volume, setting the metadata of a logical volume and removing a logical volume, etc.

The translation work done by the Comm Module includes: 1) Translation from the UUID to a local disk identifier. As described in Section 2, to identify a physical disk between the VSM and Agent, the VSM writes a UUID at a specific location on the disk, and this UUID is used in managing commands to identify this disk. It is the Comm Module's work to translate this UUID into an identifier which can be understood by the local machine. 2) Translation of formats: for performance consideration, the metadata transferred on the network is not suitable for the Driver Module, and the Driver Module can only receive data in a continuous memory space. It is also the Comm Module's work to do this translation.

The Comm Module works entirely in background and displays no message. For observation of the executing results, the Comm Module sends the messages back to VSM. The administrator can watch these messages through a command line or GUI.

3.2 Driver Module

When loaded, the Driver Module creates the admin pseudo device for the Comm Module and the Comm Module can use some special IOCTLs to send commands to the Driver Module through the admin pseudo device. After receiving metadata from the Comm Module, the Driver Module saves it and allocates some data structure for it such as locks or reference counts. When the metadata is updated on the VSM, the Driver Module also updates its copy according to the managing command.

The main task of the Driver Module is to perform address mapping when there is an I/O access. Our system supports two kinds of address mapping —linear and stripe. A linear volume is used to enlarge the size and a stripe volume is used to improve performance. These two can be used in a mixed way.

Address mapping and dispatching is the key process in the whole system. Solaris uses a buf_t data structure to describe an access [9]. In the Driver Module, the disk accesses are accomplished in an asynchronous way: the Driver Module receives the original buf_t structure containing the logical address from the application and breaks it into several child buf_ts containing the physical addresses. Each physical buf_t is bound with the completion routine and then sent to the lower disk driver. When an access is finished, the OS notifies the Driver Module by calling the completion routine, which means that the Driver Module does not wait for the accomplishment of these accesses. When all the child buf_ts are finished, the Driver Module notifies the upper level of the accomplishment of the parent buf_t. Only through this method could the advantages of the stripe volume be realized to improve performance.

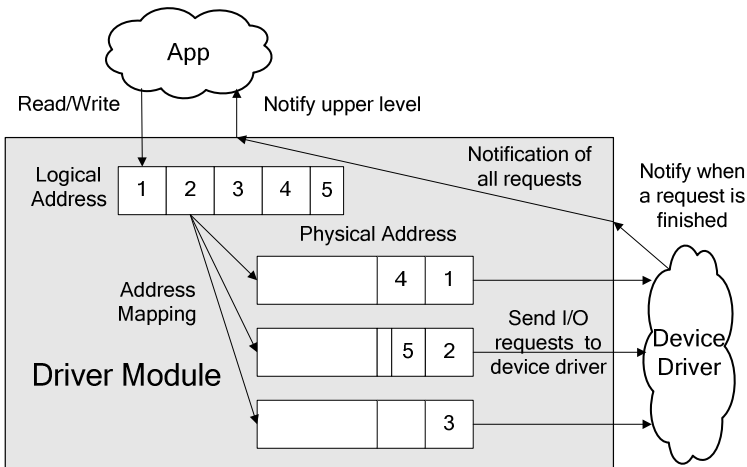


Fig. 3. Process of an I/O access

Fig 3 describes the process of an I/O access to a stripe volume consisting of three physical disks. At first, the application gives the logical address to access, and the Driver Module performs the address mapping. In this example, the whole logical address was mapped to five segments of physical addresses on three disks. The Driver Module sends these requests (buf_ts) to a lower disk driver for asynchronous I/O

access. Each time the disk driver finishes a request, it will notify the Driver Module. After the Driver Module has been notified of all the five requests, it returns the result to the application. The access from the Driver Module to the lower disk driver is asynchronous, but the access from the application to the Driver Module can be either asynchronous or synchronous, according to the requirement of the application.

4 Design and Implementation of On-Line Extension

On-line extension of a logical volume includes two steps: the update of metadata and the refreshment of the file system. The former is quite simple: the VSM sends the new metadata to the Agent through their Comm Modules. But the latter is quite complex. The file system level is higher than disk device level logically. Thus, the refreshment of the file system should not be implemented in the device driver, but should be done by the OS. So the extension of a logical volume requires support from both the virtualization system and the OS and its file system. Solaris 10 can achieve this job on the UFS (growfs) [10]. In the process of extension, synchronization operations such as lock and unlock are necessary. The locking time should not be too long for consideration of the time needed to respond to an application. Three types of extension processes are discussed as follows:

1) Logical Volume without a file system: Some database software can bypass the file system and use the block device directly. For a logical volume without a file system, the process is as follows: locking metadata in the Driver Module, updating metadata and unlocking metadata. This is a simple and fast process.

2) Logical Volume with a file system but not mounted: The refreshment of the file system requires access to the logical volume. In this process, the metadata cannot be locked. To keep data synchronization, it is necessary to instruct the OS to lock the file system first (a writing lock is sufficient), and then the refreshment of the file system can be done. Solaris provides such support (lockfs, growfs) [10].

3) Logical Volume with a file system and mounted: This is the most common status. On-line extension requires that the volume cannot be unmounted. Support from the OS is also required. The process is similar to 2). Solaris 10 provides support for this status (growfs -M). If the size of a logical volume changes greatly or the machine is slow, the refreshment of the file system takes a long time. In the process, the volume is locked and all I/O accesses are suspended. To solve this problem, a logical volume can be enlarged in several steps: after the file system is locked, only part of the file system is refreshed and then the file system is unlocked. This process is repeated several times until the whole refreshment is accomplished. In this way, the response time to applications is reduced. Fig.4 describes this process.

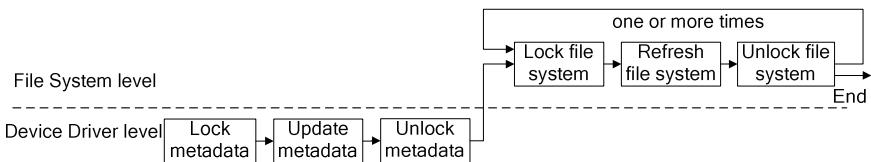


Fig. 4. Extension of a logical volume with a file system and mounted

5 Results and Analysis

We used Iometer [11] to test the performance of the out-of-band virtualization system. In the test, a Solaris Agent, a VSM and an FC disk array were used. The Solaris Agent was installed on a two-way 1.2 GHz UltraSPARC-III Cu machine with 4GB of memory and a Qlogic 2310 HBA card running SunOS Release 5.10 Version. The VSM was installed on a two-way 2.4 GHz Intel Xeon machine with 1 GB of memory and an Emulex LP982 HBA card running Linux kernel v.2.4.16. Via a Brocade Silk Worm 3800 fibre channel switch, these machines were connected with an FC disk array controlling five 146 GB Seagate Cheetah 10K disks.

The parameters of the Iometer were as follows: 4 worker threads; 80% reading access and 100% random access; the size of the logical volume (linear and stripe) was 8GB; every test lasted for 15 minutes. The file system used by Solaris was UFS. The size of I/O requests ranged from 8KB to 4MB. The stripe volume consisted of 4 FC disks with a stripe size of 64KB.

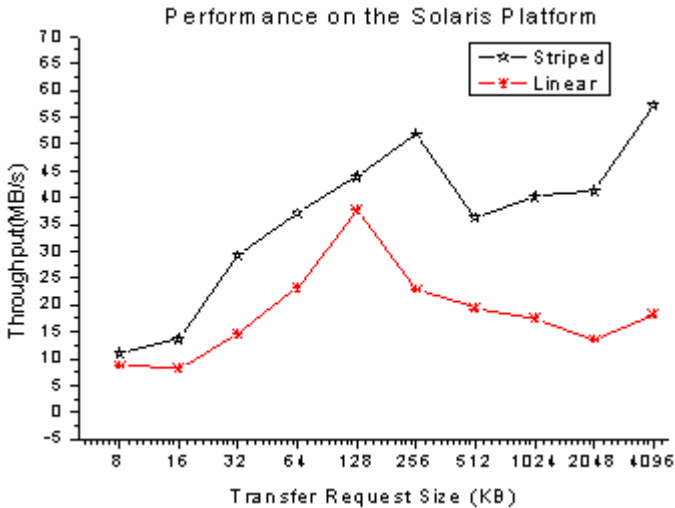


Fig. 5. Results of the performance test

The results of the performance test are shown in Fig 5. The stripe volume exceeded the linear volume to different extents. The stripe volume consisting of 4 disks exceeded the linear volume by a maximum of 215.01%, a minimum of 16.77% and an average of 103.65%. The advantage of a stripe volume is not distinct in small-size requests as a result of cache and file system influence, but it is quite significant in large-size requests when the volume performance becomes the key factor. At the last two points (2048KB and 4096KB), the throughputs of the stripe volume are both more than 3 times than those of the linear volume. The figure depicts the performance in a real system with different request sizes. The right part of the figure shows the performance of the logical volume itself.

6 Conclusion

This article introduces the design and implementation of an improved out-of-band virtualization system on Solaris 10. This system distributes the overhead of address mapping to every Agent, improving performance, reliability and scalability. It also implements on-line extension of a logical volume, providing better support for 7×24 uninterrupted service.

A performance test with Iometer on UFS showed that the performance of a stripe volume consisting of four disks exceeded that of a linear volume by an average of 103.65%. The advantage of the stripe volume was distinct when the request size was large, providing more than three times of throughput than the linear volume.

References

1. The Storage Networking Industry Association (SNIA). Storage Virtualization I: What, Why, Where and How.
2. André Brinkmann, Michael Heidebuer, Friedhelm Meyer auf der Heide, Ulrich Rückert, Kay Salzwedel, and Mario Vodisek. V:Drive—Costs and Benefits of an Out-of-Band Storage Virtualization System, In Proceedings of the 12th NASA Goddard, 21st IEEE Conference on Mass Storage Systems and Technologies (MSST), College Park, Maryland, USA, 13 - 16 April 2004.
3. Shu Ji-wu, Li Bigang, Zheng Wei-min, Design and Implementation of a SAN System Based on the Fiber Channel Protocol, IEEE Transactions on Computers, 54(4), 2005: 439-448.
4. Spinnaker Networks. White Paper: A New Approach to Storage Virtualization.2002.
5. Meng Ran, Shu Ji-wu, Xue Wei, Design and Implementation of an Out-of-Band SAN virtualization system Based on Windows NT Architecture, IFIP International Conference on Network and Parallel Computing 2005, LNCS 3779,371-378.
6. Zhang Guanyin, Shu Jiwu, Xue wei, Zheng Weimin, MagicStore: A New Out-of-Band Storage Virtualization System in SAN Environment, IFIP International Conference on Network and Parallel Computing 2005, LNCS 3779,379-386.
7. HP Open View Storage Operations Manager, <http://h18006.www1.hp.com/products/storage/software/som/index.html>.
8. Charles Milligan, Sid Selkirk. Online Storage Virtualization: The key to managing the data explosion, Proceedings of the 35th Hawaii International Conference on System Sciences – 2002.
9. Sun Microsystems, Inc. Writing Device Drivers. Part No: 816–4854–10. Jan. 2005. <http://docs-pdf.sun.com/816-4854/816-4854.pdf>.
10. Sun Microsystems, Inc. Solaris Volume Manager Administration Guide. Part No: 816–4520–10. Jan 2005. <http://docs-pdf.sun.com/816-4520/816-4520.pdf>.
11. IoMeter Project, <http://sourceforge.net/projects/iometer/>.

High Performance Virtual Backup and Archive System*

Dan Feng, Lingfang Zeng, Fang Wang, and Peng Xia

Key Laboratory of Data Storage System, Ministry of Education
School of Computer, Huazhong University of Science and Technology, Wuhan, China
dfeng@hust.edu.cn, zenglingfang@tom.com

Abstract. Built on a sequential write/read device, a tape library is seldom considered as a viable place for fast backup/restore data. With the help of the virtualization technology, in this paper we propose a virtual backup and archive system, called VBAS. The purpose of VBAS is to maintain a consistent view of mass storage so that the user can effectively manage it. And VBAS allows users to create files and directories as well as delete, open, close, read, write and/or extend the files on the device(s). VBAS maintains security on the files and provides the management for fragmentation. Moreover, VBAS can support large-scale file systems. Users have two ways to access VBAS: using general backup application, and through the APIs provided by VBAS. Based on VTL, RAID-DP, and iSCSI, VBAS not only has the disk-file-system-like functions, but also retains the characteristics of tape library storage, thus achieving a good tradeoff between cost and performance. The prototype system performance is presented and improvements are analyzed to achieve higher write/read performance.

1 Introduction

Tape is by far the most popular media for the near-line or offline (e.g. archiving data) storage. As the amount of stored data in many data-intensive applications, such as high energy physics research, weather prediction, spatial data and seismic data analysis, increases dramatically, tape libraries can play an important role in maintaining and backing up critical data. The disadvantage of tape drives is that they are sequential-access devices. This makes them much too slow for general-purpose storage operations.

The primary driving force behind users' adoption of archiving has traditionally been to reduce total cost of ownership (TCO). Though cost is often thought to be the most important factor, it ranks lower than data retrieval performance and scalability in the selection criteria for an archiving solution. Users want to achieve file retrieval in minutes or even seconds.

* This paper is supported at Huazhong University of Science and Technology by the National Basic Research Program of China (973 Program) under Grant No. 2004CB318201, National Science Foundation of China No.60273074, No.60303032, Huo Yingdong Education Foundation No.91068.

Nowadays, many file systems based on disk and tape storages have been presented. AMASS [1] for UNIX, designed for direct-attached and network-attached storage environments, is a cost-effective solution for enterprises that have more data than their disk capacity can support. AMASS transforms libraries into truly online direct-access mass storage. AMASS presents an automated optical, tape, or DVD library as one device and one mount point (showed in Figure 1.), through a standard UNIX file system interface consisting of directories and files. Users or applications can save or access files residing on libraries in the same way they would with magnetic disk. AMASS provides transparent access to data on disk and tape library storage.

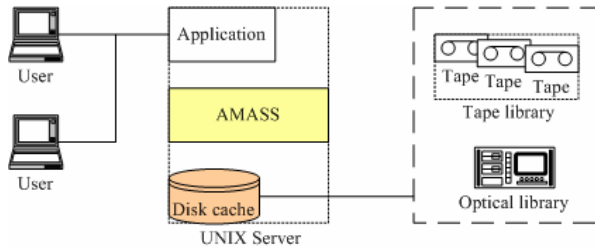


Fig. 1. AMASS provides transparent access to data on disk and library storage

However, disks in AMASS are used based on the general-purpose file system, and thus they have to confront the same set of problems inherited from general-purpose file systems.

SEPATON's Disk Dynamic File System allows large I/O streams to execute efficiently and has the built-in infrastructure to dynamically balance performance across all available disks in their VTL appliance. The Disk Dynamic File System has the important side effects of not only sustaining maximum throughputs, but also dynamically load balancing I/O streams without any requirement for performance “tuning” [2]. But, The Disk Dynamic File System is still a disk-related file system and it is not feasible for the management of tape library storage system.

Our virtual backup and archive system (VBAS) effectively integrates the virtual tape library (VTL) [3], the VTS technology [10], the RAID-DP technology [4], and the iSCSI [5] technology to provide transparent tape file access for users while retaining other functions of tape libraries. Our study of VBAS shows the main advantages of VBAS over the conventional tape libraries and conventional (disk-based) file systems as follows: (1) High backup and restore performance with disk-file-system-like functions. (2) Low cost compared with simple disk-based system. (3) Some finer functions integrated both RAID and the tape library.

The main contributions of this paper are: (1) Overcomes general file systems more prone to being infested with viruses and having an inherent problem of fragmentation. (2) Provides a virtual backup and archive system based on RAID. (3) Implements the prototype system via virtual tape library technology and presents and discusses the experiment results.

The rest of the paper is organized as follows. The design and implementation of VBAS is presented in the section that follows. Test results and performance analysis

are discussed in Section 3. Finally, we give the conclusions and outline some remaining problems in Section 4.

2 Design and Implementation of VBAS

VBAS enables data management without disrupting end-user or application accesses. Done correctly, the VBAS can reduce storage management overhead, simplify end-user accesses, and enable additional storage management functionalities. The VBAS solutions transform a mass storage system from a “two-size-fits-all” (high-performance disk and low-cost tape) world into a sophisticated mix of discrete storage elements with a wide variety of performance, cost and capacity attributes. The alphabet soup of technologies (VTL, iSCSI, SATA, etc.) presents an excellent opportunity for the storage systems applications to locate the right data on the right device at the right time.

2.1 Hardware Architecture of a VBAS-Based Testbed

Figure 2 shows the hardware architecture of our mass storage system testbed based on VBAS. The target device comprises a RAID and a tape library connected by a SCSI channel, also called a hybrid device. The console and application servers (Web server, E-mail server etc.) and the target device are interconnected by a TCP/IP network. The console, web server, e-mail server and backup server form an initiator and they access data in the hybrid device through the iSCSI protocol.

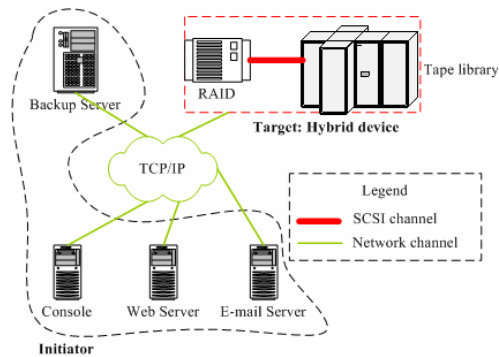


Fig. 2. Hardware architecture of the mass storage system testbed based on VBAS

2.2 The VBAS-Based Software Deployment

iSCSI builds on the two most widely used protocols from the storage and the networking worlds. From the storage side, iSCSI uses the SCSI command set, the core storage commands used throughout all storage configurations. On the networking side, iSCSI uses IP and Ethernet that form the basis for most enterprise networks and used in metropolitan and wide area networks.

With the help of iSCSI, shown in Figure 2, almost all of backup applications, such as tar [6], taper [7] and bacula [8], can access the target (hybrid device) via VBAS. It is possible that, for a backup application, the daemon servers may be deployed in the backup server while its proxies may be installed in the web server or E-mail server.

Figure 3 shows the functional and logical relationship among the function modules in the initiator and the target. The target device refers to an entity that presents itself as a SCSI direct access disk and tape sequential access while running within the Linux kernel space. In the user space implementation, the entity providing the SCSI functionality and the entity responsible for transmitting SCSI over a given protocol form one logical piece of code. However, in the kernel space implementation, the entity responsible for handling SCSI commands has an existence independent of the low-level front-end target driver that is responsible for the transmission of SCSI in a device-specific manner. Thus, in terms of visualization, it may be better to think of the kernel space modules as consisting of two distinct entities - the generic SCSI target mid-level and the low-level front-end protocol-specific target driver. The VBAS module is implemented in the initiator. It can work with the traditional disk-related file system and this solution is transparent for user space application. Also, users can custom their applications based on the APIs provided by VBAS.

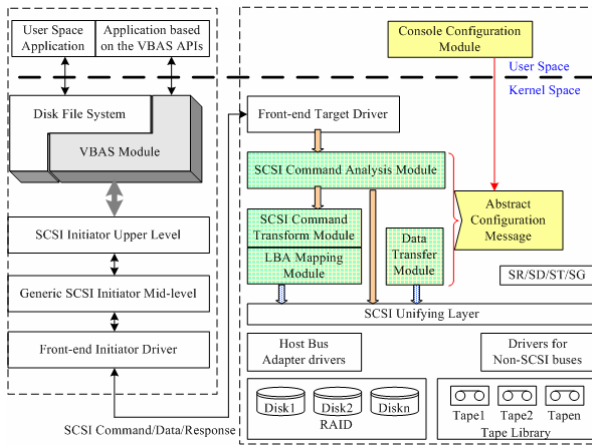


Fig. 3. The functional and logical relationship among the function modules of the prototype system

In the target, our function modules are implemented in SCST (The SCSI target mid-level subsystem for Linux) [9], which is a generic SCSI target middle level for Linux. It is designed to provide unified, consistent interface between SCSI target drivers and Linux kernel and simplify target drivers development as much as possible. Although data distribution policy is different comparing with [10] [11], in substance, their implementation technologies are analogical. And they all have to record all the logical objects [12] on the RAID. In Figure 3, the SCSI command analysis module receives SCSI sequential commands from the backup application, and determines whether the commands should be executed on the RAID or on the tape library. Then it delivers them to the proper module (or media). The SCSI command transform module

is responsible for transforming SCSI sequential commands into SCSI block commands. The LBA (logical block address) mapping module maintains the block mapping information, which associates the logical unit of an object with its logical block address in the RAID. The data transfer module performs data transfer between RAID and tape library according to some information lifecycle management policy.

In addition, VBAS provides a client application that is implemented using the API of VBAS. The client application may be deployed in the console (Figure 2.) and perform remote file management operations. Users also can implement their own remote file management application by the API of VBAS.

2.3 The VBAS Data Structures

Traditionally, a file system represents the logical structures and software routines used to control access to the storage on a hard disk system. However, VBAS, a tape-based file system that hides the details about tapes and, provides an API for applications, is designed and implemented to provide the “access by name” functions, including tape file creation, tape file read, tape update, tape file deletion, tape file copy, tape file renaming and tape defragmentation.

1. VBAS data structures and functions

The file allocation table (FAT) of every library slot and every tape are defined. The FILE_NODE constructs the file name information and the FILE_RECORD records some information of file in true tapes. The virtual tape list and logical object list are defined. For the *type* element, its value may be one of the defined constants – LOGICAL_BLOCK, FILE_MARK, SET_MARK, BEGIN_NODE or END_NODE.

Some main functions (in target software) dispose the transform from SCSI stream command to SCSI block command, such as INQUIRY (0x12), REWIND (0x01), READ (0x8), WRITE (0xA), MODE_SENSE (0x1A), WRITE_FILEMARKS (0x10), SPACE (0x11) etc.. Specially, for the RAID, the command type of write and read is 10, so the VBAS has two transform functions (transform_write_6to_10 and transform_read_6to_10).

2. The system/configure files in VBAS

At the same time, VBAS provides some system files and configure files which facilitate the configuration about VBAS. The system files record some file metadata information in VBAS, and the configure files provides the configure information of VBAS. For instance, the TapeLibrarySlotInfo.txt records the tape library slot information (defined by SLOT_FAT) both in the VTL and the true tape library. The TapeVBASFAT.txt stores the information of tape file allocation table (defined by TAPE_FAT) in VBAS. The FileName.rec gives file name information (defined by FILE_RECORD), and the configure information for the VBAS is set or got by the administrator.

3. File operation algorithms in VBAS

This subsection shows those file operation algorithms, such as create, dir, read, update, copy, erase, rename and defragment etc. For each file operation algorithm, it begins at the initialization file system function - InitFileSystem(), and ends at the close file system function – CloseFileSystem(). Because those algorithms aim at the write or read files in a tape, they are all applicable to both the true tape library and the VTL. Also, as mentioned above, in the initiator, users can also use the APIs provided by the

VBAS to implement tape file management. Key file operation algorithms for VBAS are as follows (update, copy, erase, rename and defragment are omitted):

Create {

1: Open the file TapeLibrarySlotInfo.txt in VBAS, create read/write buffer, initiate the process of read/write and open the device file (e.g. st0 or nst0) of VTL.

2: Judge if the online VTL tape runs out of space or not. If the tape runs out of space, exchange a new VTL tape.

3: Deal with the name confliction according to the SLOT_FAT and open the file of tape file allocation: TapeVBASFAT.txt, and form the current TAPE_FAT.

4: Write data to the buffer.

5: The write process waits for that the buffer becomes full, and writes data to the VTL tape.

6: Write the remainder data in the buffer and close the FileName.rec, TapeVBASFAT.txt and TapeLibrarySlotInfo.txt.

7: Free the read/write buffer. And stop read/write process and close the device file of VTL.

}

Dir {

1: Open the file TapeLibrarySlotInfo.txt in VBAS, create read/write buffer, initiate the process of read/write and open the device file of VTL.

2: Read the every item in the TAPE_FAT of VBAS and list them one by one.

3: Close the VBAS and close the device file of VTL.

}

Read {

1: Open the file TapeLibrarySlotInfo.txt in VBAS, create read/write buffer, initiate the process of read/write and open the device file of VTL.

2: Judge if the required file is in the online VTL tape or not. If not, exchange a new VTL tape, and open the files: TapeVBASFAT.txt and FileName.rec.

3: The read process reads the file to the buffer according to the first address of the TAPE_FAT and the FILE_RECORD.

4: The application reads data from the file buffer by the read function of VBAS.

5: Close the FileName.rec, TapeVBASFAT.txt and TapeLibrarySlotInfo.txt.

6: Free the read/write buffer and stop read/write process and close the device file of VTL, close VBAS.

}

3 Test Results and Performance Analysis

To test the write/read performance of VBAS, Windows 2000 and Redhat Linux were used in front-end servers. Under Windows 2000, we adopted Auto Backup. Under the Redhat Linux system (Linux kernel 2.4.20-8), we adopted tar and Taper [7] as the backup software. These three kinds of backup software performed well with VBAS. The results indicate that VBAS has compatibility with multiple operating systems and backup software.

We tested the performance of VBAS by using taper under Redhat Linux (also Linux kernel 2.4.20-8). The tape drive we adopted was HP MSL5030, and the tape media was

hp ultrium 200GB data cartridge (C7971A). We adopted the SEGATE ST3404LC SCSI disk to simulate a VTL tape.

Our main concern is the backup time (write performance) and the restore time (read performance) for different primary backup devices. Also, the write/read performance both in different Ethernet and in the local node was tested in our experiment. Table 1 and 2 show the results. There are four group data, and each group recorded those test results about the write time and the read time. Moreover, we adopted different number of file, and the total size of file(s) was also ranked from 50Mbyte to 1Gbyte.

Table 1. Throughput in different Ethernet

Number of file	Total size of file (Mbyte)	The VTL tape in VBAS			
		1000M Ethernet		100M Ethernet	
		Write Throughput (Mbyte/Min)	Read Throughput (Mbyte/Min)	Write Throughput (Mbyte/Min)	Read Throughput (Mbyte/Min)
1	50.0	1509.7	1000.0	603.8	750.0
1	200.0	1182.0	705.9	1006.4	571.4
1	1000.0	1115.1	714.3	1090.0	566.0
381	85.0	850.6	1020	850.6	850.6
3386	455.1	684.7	941.6	681.2	910.2
22216	1000.0	503.9	759.5	503.9	659.3

Table 2. The throughput comparison of the local node of VTL and the physical tape

Number of file	Total size of file (Mbyte)	The VTL in Local Node		The Physical Tape	
		Write Throughput (Mbyte/Min)	Read Throughput (Mbyte/Min)	Write Throughput (Mbyte/Min)	Read Throughput (Mbyte/Min)
1	50.0	1509.7	1509.7	377.4	187.5
1	200.0	1313.3	750.0	407.5	203.4
1	1000	1257.4	750.0	399.3	196.7
381	85.0	1020.7	1275.0	510.3	212.5
3386	455.1	989.1	1335.3	460.4	224.4
22216	1000.0	672.1	1153.8	427.7	269.1

The results showed that the average write time of the VTL tape in 1000M Ethernet is 99.14% of that in 100M Ethernet and the average read time in 1000M Ethernet is 95.68% of that in 100M Ethernet, which indicated that the different network environment has few influence for VBAS. However, the test results in local node indicate that network has large influence for VBAS. For instance, the average write time of the VTL tape in local host is 54.62% of that in 100M Ethernet and the average read time in local host is 61.64% of that in 100M Ethernet.

At the same time, the size of single file much affects the write performance of VBAS. For example, Table 1 and Table 2 show the write throughput are descending with the ascending number of file. Moreover, the average backup speed of the VTL

tape in local node is 2.65 times of that of the physical tape and the average restore speed of the VTL tape in local node is 3.77 times of that of the physical tape, which indicate that the VTL tape in VBAS can enhance the backup speed greatly. So we can get a conclusion that our VBAS is fit for backup and restore applications, specially, for large size file.

4 Conclusion and the Future Work

It is important to note that simply replacing tape with low-cost disk will not provide the technological advantages. A disk-based backup solution can provide for smaller backup windows, and also provide somewhat faster recovery of distinct data files. But again, a disk-based solution does not provide off-site protection. VBAS significantly improves backup and restore by enabling all the data to remain online for faster, consistent restores. VBAS opens doors for new strategic applications by removing the cost and complexity of large quantities of traditional disk storage. The other traditional file system problems regarding performance and security are also no longer a concern. This is due to the fact that VBAS is typically proprietary and designed to act similar to a tape system with performance of disks. So, it is the primary benefits from archiving for end users to reduce in primary disk space and to improve performance (as expressed in faster response times).

References

1. Website, May, 2005, <http://www.adic.com/>
2. Paul Feresten. Comparing Host-Based D2D to VTLs for Backup and Restore - Part 2. Website, 2005. http://www.wvpi.com/index.php?option=com_content&task=view&id=132&Itemid=67
3. T. E. Anderson, M. D. Dahlin, J. M. Neefe, D. A. Patterson, D. S. Roselli, and R. Y. Wang. Serverless network file systems. *ACM Transactions on Computer Systems*, 14(1):41–79, February 1996.
4. Chris Lueth, Network Appliance, Inc. October, 2004. NetApp Data Protection: Double Parity RAID for Enhanced Data Protection with RAID-DP. March 5, 2005, available from: http://www.netapp.com/tech_library/3298.html
5. J. Satran et al. Internet Small Computer Systems Interface (iSCSI). Available from: <http://www.ietf.org/rfc/rfc3720.txt>, April 2004.
6. Website, January 10, 2005, <http://savannah.gnu.org/projects/tar/>
7. Website, January 10, 2005, <http://www.e-survey.net.au/taper/>
8. Website, January 10, 2005, http://www.linux.org/apps/AppId_8816.html
9. Ashish A. Palekar etc. Design and Implementation of a Linux SCSI Target for Storage Area Networks. Proceedings of the 5th Annual Linux Showcase & Conference, 2001.
10. Mu Fei, SHU Ji-wu, Li Bigang, ZHENG Wei-min. A Virtual Tape System Based on Storage Area Networks. H. Jin, Y Pan, N. Xiao and J. Sun (Eds.), GCC'2004 Workshop on Storage Grid and Technologies, LNCS 3252, pp.278-285, 2004.
11. Jussi Myllymaki, Miron Livny. Disk-tape joins: synchronizing disk and tape access. *ACM SIGMETRICS Performance Evaluation Review*, pp.279 – 290, 1995.
12. ANSI, SCSI Stream Commands-2 (SSC-2), revision 09, 9 July 2003, <http://www.t10.org>

Insurable Storage Services: Creating a Marketplace for Long-Term Document Archival

Rahul Simha¹ and K. Gopinath²

¹ Department of Computer Science
The George Washington University, Washington, DC 20052, USA

² Department of Computer Science and Automation
Indian Institute of Science, Bangalore, India
simha@gwu.edu, gopi@csa.iisc.ernet.in

Abstract. Digital storage is a key element not only of computing systems, but is now considered an essential component of the infrastructure of any modern organization. This need has co-evolved with the technology that has grown rapidly in recent years to provide low-cost high-capacity storage. At the same time, the storage needs of users have now become more sophisticated and diverse. Some users require very long-term preservation; others need high security; and still others ask for highly-reliable, distributed storage solutions. These needs pose a problem for solution providers in that no single solution seems to meet all needs. Similarly, users must construct services out of disk systems on their own. This paper proposes a way to streamline the marketplace through *insurable storage services*, a combination of two ideas. The first is to define different categories of storage service; the assumption here is that a refined categorization will better identify particular user needs. The second, and more substantive idea, is to treat digital documents as insurable property. The insurance of storage will provide economic incentives for both producers (storage service providers) and consumers (individuals, organizations) to jointly create a marketplace that provides a diversity of differentially-priced services. For example, insurers can help assess the durability of storage solutions and provide consumers with a quantitative valuation (“It’ll cost you \$x per GB to ensure that your documents last 100 years”). Similarly, storage service providers will have incentives to maintain multiple geographically distributed copies, and to continually move the copies onto emerging technologies (“You’ll need to store more copies if you want a higher reliability rating”).

1 Introduction

Digital storage has grown from being a mere “part of a computer” to the complex array of interacting technologies, file systems, services and service providers that defines the world of storage today. This world now has evolved into subspecialties such as materials, solid-state memories, disk-systems, RAID, storage area networks and storage services, to name a few.

In this paper, we identify some possibilities for future storage services, and most importantly, focus on a market mechanism by which these services could

arise and grow naturally. The lack of such a dynamic marketplace is one of several key problems identified in the recent report on digital preservation [10]. In the words of that report, “Creating an economy for long-term preservation entails providing incentives for organizations to invest in digital archives, even though some of the benefits of investments made today may not be realized for decades.”

Our approach to addressing this problem starts with the observation that most current storage-service paradigms posit two players in the service market: the user and the Storage Service Provider (SSP). By introducing a third player, insurers, and by treating digital documents as property, we argue that a natural marketplace will emerge as the insured and the insurers drive the creation of new, cost-effective services. This way, investments can be funded today for the benefits realized later. The purpose of this paper is to provide an overview of this idea, to identify factors affecting insurance and to raise a few new technical issues.

2 Background

2.1 System Components

As background, let us outline the various components and players in a storage service system. Consider a typical desktop user who is busy creating, updating and deleting files. This creates a stream of I/O between processor and disk. For many users, the local disk is the final resting place for documents instead of being merely a local cache of a more stable repository. In today’s computing paradigm, individual users either perform their own backups or are part of a networked backup system. In either case, files are scheduled for backup on a periodic basis. When the individual user is part of a larger organization, the organization may contract with a SSP to perform such a backup, or even store the primary copy on site.

In our paradigm, we identify the following players. First, there are *users*, either as individuals or as part of an organization. Each user machine runs software to feed files into a pipe intended for storage. That is, the operating system, instead of writing to a local disk hands over the file to an application software running on behalf of a service provider. This application will use the local disk as a cache for efficiency. Second, *SSP’s* at the other end of this pipe provide a variety of storage services, including storing the primary file and several copies. Finally, some services will need third-party providers. These include: trusted third-parties for some types of security operations (we will see one example later in Section 4.1), third-party services for search, or a third-party curation service aimed at long-term archival for maintaining a historical record.

2.2 User Needs

What do users want out of a storage provider? Let us first consider individual users in the home or small-business environment:

- *Backup.* Clearly, this most common need today will continue into the future. In addition to simple backup, different users may desire various levels of quality. For example, small business users might opt for multiple, geographically diverse backup copies.
- *Legacy.* Users will want to pass on their digital property to rightful heirs and ensure their access. Some of these heirs may be family members, others might include a trust or a public organization. Today, this type of service is virtually non-existent and certainly not systematically addressed.
- *Legal services.* As with any kind of property, in case of conflict over a legal document, storage providers should make available a framework for conflict resolution between all those party to the document. For example, a real-estate property deed will need to have its integrity assured, and if needed, to be checked against the same document stored by the local county office¹. Thus, a SSP must provide the a means for access by other interested parties, including a court of law.

In addition to these services, organizations have other requirements:

- *Strong backup and availability.* Organizations usually place greater value on the accessibility and reliability of backups than do individuals. These additional factors are influenced by the physical durability of storage warehouses, their geographical diversity, diversity across disk manufacturers, number of such diverse copies, administrative convenience and speed of archival.
- *Group coordination.* Organizations also need to store documents in collections and from groups of individuals, often removing association with individuals. For this purpose, an important service is to provide organizations with a framework for defining groups and moving documents between groups. Organizations also share documents between divisions and across other organizations.
- *Ownership.* The question of ownership is important to an organization. When documents are associated with groups or roles, individuals in those groups or roles assume temporary ownership. Thus, a framework for ownership must provide for associations between individuals and groups, and between individuals and administrative roles with high privileges.

The needs of public organizations or of government agencies also include custodial actions. For example, U.S. law dictates that the executive branch must make available its digital record to the National Archives. Even if the law does not require it, other agencies might wish to make their documents available to the public after a period of use.

2.3 Services

Each need identified above corresponds to a service that an SSP should provide. In addition to these, we identify a few more:

¹ Although cryptographic techniques can be used to verify integrity, these techniques are subject to implementation errors. Furthermore, challenges must nonetheless be addressed by third parties.

- *Security*. SSP's should go beyond simple encryption to provide a comprehensive solution to various security needs. These include key management, key transfers between groups, long-term key storage, re-keying, integrity checking, audit trails, conflict resolution and a framework for access privileges in a multi-user environment.
- *Escrow*. SSP's should provide services that aim for fixed-term secure storage, after which documents become public or are made available to other parties according to a policy. Escrow services could also be used in various legal transactions, such as on-line contract negotiations.
- *Notification services*. SSP's should provide periodic notification, just like a bank statement, that contain storage transaction histories, reports on integrity checks, updates of technologies, numbers of copies, the geographic sites where they are stored and usage statistics.
- *Search*. SSP's should provide access points for search providers so that sophisticated search technologies may be used by organizations to search within their document troves.
- *Meta information*. SSP's should provide API's for operating systems and applications so that they may store context information along with documents. This context information should be augmented with usage statistics, locality statistics (time, date) to form a comprehensive record of meta information that will be useful for later searches.

Taken together, these features point to a future of value-added services in the area of storage. Why hasn't this future emerged already? Every change in the marketplace needs a driver, a force that facilitates transactions and growth and propels providers to competitively offer creative new services. We argue that one such possible market driver is insurance.

3 Insured Storage

3.1 How It Would Work

Users and organizations today are faced with a bewildering number of options in constructing solutions for their storage needs. They must identify vendors of storage systems and then build services on top of these storage systems. Furthermore, they have no easy way to quantify the value of their investment.

Instead, we propose that users treat their digital documents as property, whose storage is insurable. To illustrate, let us consider a user interested in contracting out storage services from SSP's. The user obtains an insurance rating for various service options across vendors, and together with the prices for each, makes a decision. Thus, for example, Insurance Company *A* is willing to insure service *B* offered by SSP *C* at the rate of \$10 per MB for 5 years. Faced with choices such as these, users can discern what works best for them.

Similarly, SSP's competing in this marketplace will make optimal use of technologies to acquire the best possible insurance rating and to offer differentiated services at different prices. Thus, SSP *C* can offer a long-term archival service

(50 years) but with slow access speeds at \$10 per MB and a short-term (5 years), highly efficient service for \$40 per MB. Each of these might be rated differently by insurers. To get a high insurance rating, a SSP would have to convince insurers that, for example, they are using current technologies, exploiting geographic and vendor diversity, and creating fresh copies at reasonable time intervals.

3.2 Benefits of Insurance

The chief benefit of using insurance is that it mediates between users and providers in what could be a confusing array of options and benefits. Users are given a single number (or two) by which to assess services; the trustworthiness of this number is based on the reputation of insurers and the cost of insurance. Similarly, SSP's work towards better ratings by cycling copies onto new technologies, increasing the number of copies, using different hard-drive vendors, different geographic locations for the copies, physically securing these locations from natural catastrophes and using proven engineering practices. All of these can be fitted into statistical models that quantify reliability, availability and efficiency. Insurers, with their legions of statisticians and considerable experience with reliability models, are already suited to this type of analysis. Furthermore, SSP's will be driven to providing creative new services to meet the needs of customers, as and when these get included in the overall rating.

We conjecture that an added benefit for long-term archival would be some closure to the "format wars." As is well-known in the archival literature, there is a tension between the proprietary formats (such as Microsoft Word) that users prefer to use, and the open, human-readable formats (such as XML) that, according to archivists, offer the best hope of being readable in the future. Since a proprietary format should logically receive a low 100-year rating, users will be driven towards open formats if they are to guarantee long-term storage.

Note that other mechanisms exist that could mediate between consumers and producers in the storage marketplace, as can be found in other marketplaces. For example, a softer, non-binding form of mediation is to simply provide ratings, of the form provided by consumer protection groups. However, these are not as powerful as mechanisms such as insurance that associate high cost for inaccurately rating services.

3.3 Factors Affecting Insurance

Many factors that one should use in rating the quality of a storage service have been discussed in the literature. These include the obvious ones: number of copies, geographic diversity of copies, vendor diversity, hardware reliability, copy replacement policy, hardware replacement policy and physical security. To these we add a few new ones. These are all based on the notion that a document that can't be deciphered or precisely located in the future, even if it is known to exist, is useless.

- *Formats.* As mentioned earlier, proprietary formats, especially those subject to frequent change, are probably not suited for long-term storage. Even if

these must be used, proper emulation [8, 11, 12] can mitigate some of the disadvantages, and thus should be factored into a rating. Even among open-formats, documents that use a mixture (such as a combination of HTML and JPEG) might receive a lower rating than simple documents.

- *Searchability.* With a long-term view, documents should be retrievable not just by specifying the owner but through a variety of means. A custodian of government documents, for example, should be certain that the documents will show up in a variety of search approaches.
- *Monitoring quality.* SSP's that continuously monitor their document copies for integrity should obviously receive a better rating. Monitoring should also include hardware, network connections, software patches and security in its purview.
- *Networks.* Aside from the assessment of individual sites where copies are stored, the network used to connect these sites should also be subject to evaluation. SSP's that use multiple networks with redundant connections should get a higher rating.

4 New Technical Issues

There are a host of technical issues associated with storage systems and storage services. Here, we point out two technical issues related to long-term archival that we believe are relatively new and not yet addressed in the technical literature.

4.1 Ownership Devolution

The first issue is motivated by considering those situations in which ownership of a document must be transferred. For example, an individual user might leave her digital assets to heirs upon her demise. Similarly, the executive branch of government must leave their documents to the public after their term expires. The problem is non-trivial even when these documents are not encrypted during use – there must be a way, for example, to ensure that the policy is properly enforced. In the future, however, most documents will probably be encrypted when stored using a service. In this case, how can one guarantee that the keys will be made available? What happens if the keys are lost or if an individual loses them?

To address these issues, SSP's need a framework for handling ownership devolution. To solve problems such as key escrow, trusted third parties might be needed in addition to SSP's. For example, a trustee can perform encryption en-route to storing a document; this trustee can keep the key and be required to reveal the key at some point in the future. A trust management framework might therefore become an essential part of such services. Since access control is based on identity, third party input is needed so that trust, delegation and public keys can be negotiated.

Note that with public-key cryptography, it becomes possible to deal with anonymous users as long as they have a public key: authentication and authorization are now possible with models such as SDSI/SPKI. In this case, an

issuer authorizes specific permissions to specific principals; these credentials can be signed by the issuer to avoid tampering. For example, SDSI/SPKI provides for credentials with delegation with the assumption that locally generated public keys do not collide with other locally generated public keys elsewhere. This allows exploiting “local namespaces”: any local resource controlled by a principal can be given access permissions to others by signing this grant of permission using the public key. Access control and cryptography can now be combined into a larger framework with logic for authentication/authorization and access control. However, permission-based trust management cannot authorize principals with a certain property easily. One solution to this problem is attribute-based approach: it combines RBAC and trust management. Other approaches include proof carrying authentication [1].

4.2 Universal Document ID

Currently, documents are identified by user account, directory structure and file name. However, as storage services and providers grow, users will need to transfer documents across providers and systems. Furthermore, there are many situations where documents need to be identified with groups rather than individuals. This raises the issue of proper identification of documents for long-term use. Identifiers are needed for indexing, for maintaining meta information and for tracking.

Thus, a service that provides unique universal document identifiers will be useful. At the same time, these identifiers should maintain an individual’s privacy and not be traceable back to the individual. One solution is the use of SHA2 hashes such as sha256, sha384 and sha512 algorithms. If SHA2 is strongly collision-resistant (as is currently believed), then the hash can be used as a global identifier. Usability of such hashes can be managed with a secure mapping between names and hashes that is locally maintained. The latter service could be constructed using a mechanism similar to DNS, the mechanism for internet domain names. If HMAC-SHA2 is used with the secret being between the client and the storage provider service, both origination and integrity can also be guaranteed.

One can imagine further refinements. These IDs, if properly extended, can also be used as ways of authorizing the use of documents to others (as in DRM). One possibility is the use of “capabilities.” However, undesirable information flows are possible in such systems using capabilities in the presence of Trojans [5]. This requires that some information about the intended recipient is incorporated in the capability, and thus results in a modification of the strict capability model (see [2] for an example).

5 Summary

In this paper, we have described the benefits of using insured storage as a mechanism for driving the marketplace for storage services. In addition, we have identified a few new factors affecting insurability and outlined a couple of new technical issues relating to long-term storage.

The question that naturally arises is: will users take to insuring their documents? Our view is that, except for a few very important documents, users might be unwilling to negotiate such minutiae on a per-document basis. However, we speculate that users will be willing to insure services especially if offered on an annual basis or in terms of Gigabytes, similar to some internet services. Ultimately, the services will be offered (because they are needed) and ultimately, users will find some, perhaps informal, way of assessing their value. Insurability, in addition to opening up a new possibilities for that industry, offers a way to rapidly streamline the marketplace for storage services. Already, insurance is currently being offered for costs related to compliance with open-source software standards [6].

Finally, we recognize that associating a direct cost with storage might drive users towards placing a value on documents today, resulting in the loss of some documents that could be valuable to some future historian. This is an important issue in general that is not considered in this paper.

References

1. A.W.Appel and E.W.Felten. Proof-carrying authentication. *6th ACM Conference on Computer and Communications Security*, November 1999.
2. Azagury et al A Two Layered Approach for Securing an object store network. *1st International IEEE Security in Storage Workshop (SISW 2002)*.
3. D.Bearman. Reality and chimeras in the preservation of electronic records. *DLIB Magazine*, Vol.5, No.4, 1999.
4. A.Crespo, H.Garcia-Molina. Cost-driven design for archival repositories. *Proc. 1st Joint Conference on Digital Libraries (JCDL)*. June, 2001.
5. S.Halevi, P.Karger and D.Naor. Enforcing Confinement in Distributed Storage and a cryptographic model for access control, *IBM Tech Report*, 2002.
6. M.LaMonica. Insurer launches \$10 million open-source policy. CNET News, Oct 2005. http://news.zdnet.com/2100-3513_22-5924112.html
7. N.Li, B.Grosz and J.Feigenbaum. Delegation Logic: a logic-based approach to distributed authorization, *ACM Trans. on Info. and System Security*, Feb 2003.
8. R.A.Lorie. Long-term archiving of digital information. *IBM Research Report RJ-10185*, May 2000.
9. Technical Architecture, Ver. 0.2. National Digital Information Infrastructure and Preservation Program (NDIPP) (www.digitalpreservation.gov).
10. It's About Time: Research Challenges in Digital Archiving and Long-term Preservation. NDIPP (www.digitalpreservation.gov).
11. J.Rothenberg. Ensuring the longevity of digital documents. *Scientific American*, Vol. 272(1), January 1995.
12. J.Rothenburg. An experiment in using emulation to preserve digital publications. Technical Report, *RAND-Europe*, April 2000.

Multi-dimensional Storage QoS Guarantees for an Object-Based Storage System*

Fei Mu, Jiwu Shu, Bigang Li, and Weimin Zheng

Department of Computer Science and Technology, Tsinghua University,
100084, Beijing, P.R. China
mufei@mails.tsinghua.edu.cn

Abstract. The Object-based storage is an emerging storage architecture that could easily fulfill multi-dimensional storage QoS requests. This paper focuses on providing QoS guarantees under Object storage infrastructure along the three most prevalent dimensions: capacity, bandwidth and latency through storage resource allocation and IO commands scheduling. Firstly we propose an algorithm on storage resource mapping derived from Toyoda algorithm, which achieves efficient resource utilization through consideration of the OSDs' serving ability. Secondly we propose an object commands scheduling mechanism and develop a prototype system based on the Lustre filesystem. Through adding timestamp to each object command and scheduling the command queue by final finish time, the system can efficiently fulfill the demands on latency from the front applications.

1 Introduction

As becoming more and more complicated, storage applications urgently demand Quality of Service (QoS) guarantees along multiple dimensions such as storage capacity, data bandwidth, respond latency, reliability, security, etc [1][2][3]. For example, a voice over IP service requires a short latency; a VOD service requires a large capacity and a high bandwidth while an E-mail service may require a relatively lower capacity and a lower bandwidth but a higher security. Among them, capacity, bandwidth and latency are commonly regarded as most prevalent.

Object Storage Device (OSD) [4][5] is an emerging storage technology in recent years. It offloads storage management from host operating systems to intelligent storage devices and provides an object-level storage interface in contrast to the conventional block-level storage interface. The OSD interface is focused on moving chosen low-level storage, space management, and security functions into storage devices to enable the creation of scalable self-managed, protected and heterogeneous shared storage for storage networks.

The special architecture of Object-based storage systems achieves convenience in providing storage QoS guarantees [6][7][8]. It has better scalability by separating the

* Supported by National Grand Fundamental Research 973 Program of China under Grant No. 2004CB318205.

control path from data path, which makes the storage system be able to integrate large quantities of devices to achieve a high bandwidth. Furthermore, the front applications access data from the object storage devices through an *object* level interface. Accordingly, we can use the concept of *class* to identify different storage applications and the multi-dimensional storage QoS requests can be expressed as *attributes* attached to each object. Hence, both the front clients and the end devices can be knowledgeable of what the front applications demand. In object-based storage systems, the end storage device has computational ability, which means that the object device can adjust its object command queue to fulfill the application request.

Yiping Lu [6] and Joel C.Wu [7] proposed QoS frameworks for OSD-based storage system, which showed that OSD-based storage system is an ideal platform to provide QoS guarantees. Kevin KleinOsowski provided suggestions for improving the OSD specification and its ability to communicate QoS requirements [8]. His work concentrated only on bandwidth guarantee. StoneHenge [9] is a multi-dimensional storage virtualization system, which is able to multiplex multiple virtual disks with a distinct bandwidth, capacity, and latency attributes. But this work was based on a conventional cluster environment using block-level storage devices.

The Lustre filesystem [10][11][12] runs today on many of the largest Linux clusters in the world. At the root of Lustre is the concept of object storage. It is a good platform for testing and validating OSD concepts such as storage QoS guarantees.

We propagate our research work in two steps to fulfill the QoS request along capacity, bandwidth and latency. Firstly we allocate the total storage resource for application classes according to their QoS requirement. But for an actual storage system the workload is quite complicated and only storage resource allocation can not assure the access latency. Thus we also propose a scheme on command scheduling and propagate a prototype system to provide latency guarantee based on Lustre filesystem.

2 Storage Resource Mapping Algorithm

This storage resource mapping algorithm concentrates on the dimensions of capacity and bandwidth because various disk performance requirements can be readily translated into bandwidth requirements. According to reference [13], we have

$$T_i \leq \frac{\omega + L_{max}}{B_i} + \frac{L_{max}}{B} \quad (1)$$

In equation 1, T_i is latency of application class i , ω is the data size in the waiting queue, L_{max} is the maximum size of a single command, B is the bandwidth for a certain physical device, which could all be acquired by experiment and experience. So we can translate a request on latency into that on bandwidth. We adopt the bigger one during resource mapping.

2.1 Problem Statement

Suppose there are N object devices in the storage system which provides M application classes. Each storage application has a demand on both capacity and bandwidth which is beyond the ability of a single object device. Commonly one application

should be evenly scattered into m_j parts till each part could be guaranteed by a single object device. m_j will be settle upon different policies according to different application classes. Finally there will be $K = m_1 + m_2 + \dots + m_M$ sub-applications in the storage system, and our study focuses on how does one single object serve the scattered applications when $K > N$.

We use a two-dimension vector $\mathbf{S} = \{C, B\}$ ($C > 0, B > 0$) to express a sub-application, C indicates the capacity while B indicates the bandwidth. An object device could be expressed by this vector too. In special, we use $\mathbf{O}, \mathbf{A}, \mathbf{R}$ to represent the original resource vector, allocated resource vector and reserved resource vector. We get $\mathbf{O} = \mathbf{A} + \mathbf{R}$. Then we assume that the at last object device i selects n_i sub-applications to serve and the result set is $\{\mathbf{S}_{ij}, 0 < j \leq n_i\}$. Ideally there could be:

$$\begin{cases} \bar{A}_i = \sum_{j=1}^{n_i} \bar{S}_{ij} = \bar{O}_i (i = 1, 2, \dots, N) \\ \sum_{i=1}^N n_i = \sum_{j=1}^M m_j = K \end{cases} \tag{2}$$

This is a multi-dimensional bin-packing problem and we propose a greedy algorithm derived from Toyoda algorithm [14].

We use $\langle \mathbf{A}, \mathbf{O} \rangle$ to present the included angle between vector \mathbf{A} and vector \mathbf{O} , which is assumed to be lied between $(0, \pi]$. There is:

$$I_i = \frac{\bar{A}_i \cdot \bar{O}_i}{\|\bar{A}_i\| \cdot \|\bar{O}_i\|} = \cos \langle \bar{A}_i, \bar{O}_i \rangle \tag{3}$$

2.2 Storage Resource Mapping Principle

The resource mapping should adapt to the devices' serving ability. In virtualization work, the physical devices are always logically divided into storage slices for different virtual storage applications. The resource vector of a single slice has the same direction as that of the whole storage device. So we should minimize the included angle between the allocated resource vector and the original resource vector. Through equation 3, our goal is to maximize I_i . Nowadays many storage systems adopt replicas to enhance reliability and performance, so the resource utilization needn't be strictly maximized.

We concentrate on the $\mathbf{O}, \mathbf{A}, \mathbf{R}$ of a certain object device. We calculate I for every received sub-application. If $\|\mathbf{A} + \mathbf{S}\| < \|\mathbf{R} - \mathbf{S}\|$, then $\langle \mathbf{O}, \mathbf{A} + \mathbf{S} \rangle$ is larger than $\langle \mathbf{R} - \mathbf{S}, \mathbf{O} \rangle$, we calculate I by equation 4; while if $\|\mathbf{A} + \mathbf{S}\| > \|\mathbf{R} - \mathbf{S}\|$, $\langle \mathbf{O}, \mathbf{A} + \mathbf{S} \rangle$ is smaller than $\langle \mathbf{R} - \mathbf{S}, \mathbf{O} \rangle$, we calculate I by equation 5. Finally, the sub-application which has the largest I should be chosen to be mapped on the certain object device.

$$I = \frac{(\bar{R} - \bar{S}) \cdot \bar{O}}{\|\bar{R} - \bar{S}\| \cdot \|\bar{O}\|} \tag{4}$$

$$I = \frac{(\bar{A} + \bar{S}) \cdot \bar{O}}{\|\bar{A} + \bar{S}\| \cdot \|\bar{O}\|} \tag{5}$$

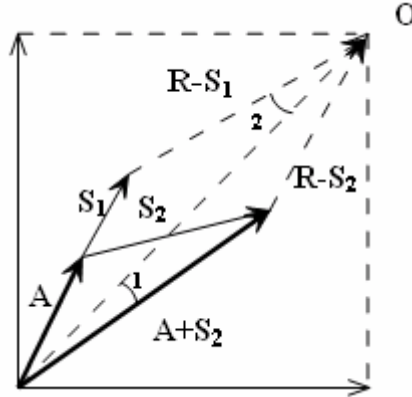


Fig. 1. $\|A+S_1\| < \|R-S_1\|$, then $\angle 2$ is smaller than $\langle A+S_1, O \rangle$; $\|A+S_2\| > \|R-S_2\|$, then $\angle 1$ is smaller than $\langle O, R-S_2 \rangle$. $\angle 1$ is smaller than $\angle 2$, so at last we choose S_2 .

2.3 Algorithm Description

The single object device allocation algorithm could be described as below:

- $\{S_j\}$: the sub-application set.
- $\{O_i\}$: the object device set.
- N : the number of object devices.
- O_i : the original resource vector of the object device i .
- A_i : the allocated resource vector of the object device i .
- R_i : the left resource vector of the object device i .

```

While  $\{S_j\} \neq \emptyset$  or there exists a available object device
  For each available element in  $\{O_i\}$ 
    For each elements in  $\{S_j\}$ 
      {
        If  $R_i$  couldn't guarantee  $S$ , continue;
        For a certain  $S$ , calculate  $I$  by equation 4 and 5.
      }
    If all  $S_j$  can not be served by  $O_i$ , mark  $O_i$  as unavailable, continue.
    Select the  $S$  for object device  $i$  according to chapter2.2.
     $A_i := A_i + S$ ,  $R_i := R_i - S$ ,  $\{S_j\} = \{S_j\} - \{S\}$ 
  }

```

If the OSD mapping process ends when none OSD is available, new object devices should be added into the OSD storage system.

2.4 Simulation Results

We generate N random object device vectors $\{O\}$, N^2 random sub-application vectors $\{S\}$ and run the mapping the OSD mapping algorithm for 500 times to get an average.

Table 1 indicates that our algorithm achieve better adaptation to physical device's serving ability than Toyoda Weighted algorithm when achieving a good resource utilization.

Table 1. Simulation results1

	N	5	10	50	100
$\langle A, O \rangle$	The algorithm	5.516	3.633	1.580	1.174
(degree)	presented here				
	Toyoda Weighted	5.737	3.713	1.812	1.293
	algorithm				
	resource utilization	74.65%	83.90%	94.17%	96.30%
	(when resource is used up)				

3 Command Scheduling Scheme

In this chapter we focus on command scheduling to provide access latency guarantee for a front application under Lustre filesystem.

3.1 Latency Guarantee System Architecture

The latency guarantee system is designed and implemented under the Lustre filesystem, which is consisted of three parts: Clients, Metadata Servers (MDS) and Object Storage Targets (OST). Lustre clients run the Lustre filesystem and interact with OSTs for file data I/O and with MDS for namespace operations.

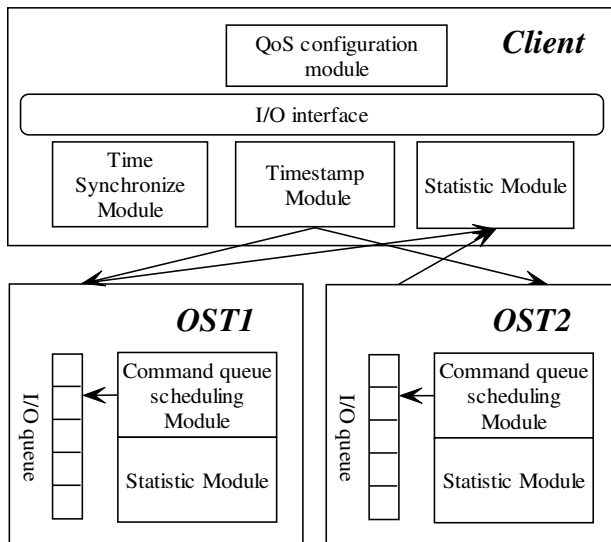


Fig. 2. The architecture of latency guarantee system

As shown in figure 2, the latency guarantee system is made up of four main components. The timestamp module receives the front object commands, recording their arrival time. It also responds for retrieving latency attributes from the objects according to various kinds of operations and adding which to the object command request. The command queue scheduling module adjusts the command queue through the information of the object command provided by the timestamp module. The statistic module will concentrate on the actual latency of the certain application and interact with the QoS configuration module. The QoS configuration module can set important parameter of the system such as common latency, coefficient for a given latency request, etc. It could indicate if the system resource is sufficient, and if the latency attributes specified by users are appropriate.

3.2 Key Technologies

Extended object commands. Between Lustre clients and the object storage targets the basic communication unit *struct ptlrpc_request* contains Lustre request message and Lustre reply message, which are all of the type *struct lustre_msg*. It is the most fundamental unit of the Lustre network protocol. This data structure is logically associated with the *object command*. In our implementation, the latency information is integrated into the *lustre_msg* while the arrival time of a certain command is integrated into the *ptlrpc_request*.

Command queue scheduling. In current Lustre system, the new arrived command request is directly added to the tail of the command queue. For a certain command request, its finish time should not be later than T^F :

$$T^F = T_{arrival} + T_{latency} \quad (6)$$

Because $T_{arrival}$ and $T_{latency}$ could be acquired precisely, we use T^F to adjust the command queue. For the commands without QoS demand, we assume their $T_{latency}$ to be the longest $T_{latency}$ required in the whole storage system. This value could be modified according the statistic information. We calculate T^F when receiving a new command, then search along the command queue from the tail to find the first command which has a smaller T^F and insert the new coming command behind it. During this process, if we meet a command having the same IO object ID with the new arrival command, then just insert the new command right behind it. Hence, the commands in the queue are nearly sorted by their T^F . The sequence of the commands which come from the same application or access the same destination has been maintained too.

3.3 Testing Results

In our testing environment, the client, MDS and OST are located in the same server. We run two applications with high workload. We give a shorter latency guarantee to application 1, while a longer one to application 2. Our statistics module counted that the latency guarantee ratio for application 1 and application 2 are 95.6% and 99.1%.

Experiment results show that the quantity of command requests is approximately linear to the total recommended data buffer size, so we use IOPS to evaluate the effect

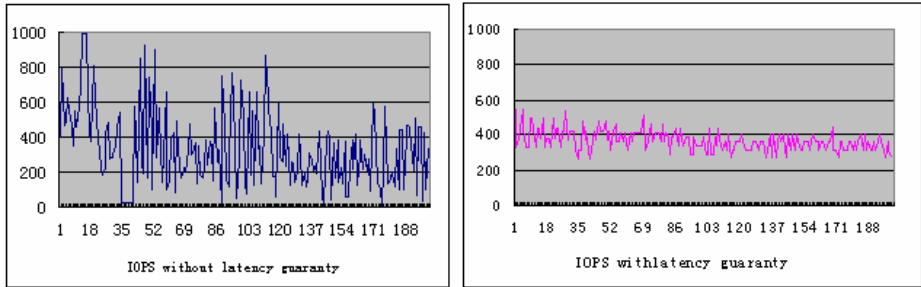


Fig. 3. Testing results of the latency guaranty system. The x axis presents the seconds and the y axis presents the IOPS.

of latency guarantee system. We focus on application 1, its IOPS without latency guarantee system and that with latency guarantee system are showed as Fig 3.

4 Conclusion and Future Work

This paper aims at providing multi-dimensional storage QoS guarantees for an Object Storage System. We focus on the three most prevalent dimensions: capacity, bandwidth and latency. We propose a storage resource allocation algorithm based on the principle of adapting to physical devices' serving ability when achieving efficient storage resource utilization. Considering the complicated application workload, we also propose a command scheduling scheme under Lustre filesystem. Testing result from the prototype system shows that the command scheduling scheme is impactful.

We will go on to study the load balancing between object devices. Command scheduling within the object devices is also challengeable. We will also concern about the emerging storage QoS dimension such as reliability, security, etc.

References

1. C. R. Lumb, A. Mrchant, G. A. Alvarez: Façade: virtual storage devices with performance guarantees, In Conference on File and Storage Technology (FAST 03), 2003
2. Z. Dimitrijevic and R. Rangaswami: Quality of service support for real-time storage systems, in Proc. of Intl. IPSI-2003 Conference, 2003.
3. M. de Miguel, J. Ruiz, and M. Garcia. QoS-Aware Component Frameworks. In Tenth IEEE International Workshop on Quality of Service, pages 161-169, May 2002.
4. M. Mesnier, G. R. Ganger, E. Riedel: Object-Based Storage. IEEE Communications Magazine, Vol. 41, No.8, pp 84-90, August 2003
5. R. O. Webster. Information Technology - SCSI Object-Based Storage Device Commands (OSD). February 2004. Rev 9.
6. Yingping Lu, David H.C. Du, Tom Ruwart: QoS Provisioning Framework for an OSD-based Storage System. NASA Goddard Conference on Mass Storage Systems and Technologies (MSST'05), 2005

7. Joel Wu and Scott A. Brandt, "QoS Support in Object-based Storage Devices," International Workshop on Storage Network Architecture and Parallel I/O (SNAPI 05), held in conjunction with the International Conference on Parallel Architectures and Compilation Techniques (PACT 2005), Saint Louis, Missouri, September 17–21, 2005
8. Kevin KleinOowski, Tom Ruwart, David J. Lilja: Communicating Quality of Service Requirements to an Object-Based Storage Device. NASA Goddard Conference on Mass Storage Systems and Technologies (MSST'05), 2005
9. Lan Huang, Gang Peng, and Tzi-cker Chiueh. Multi-dimensional storage virtualization. SIGMETRICS Perform. Eval. Rev., 32(1), 2004, 14~24
10. P. Schwan. Lustre: Building a file system for 1000-node clusters. In Proceedings of the 2003 Linux Symposium, July 2003
11. Lustre project, <http://www.lustre.org>
12. P. J. Braam. The Lustre storage architecture, 2002
13. A. K. Parekh, R. G. Gallagher. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. IEEE/ACM Transactions on Networking, 2(2), 1994, 137~150
14. Y. Toyoda: A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Management Science, 21(12):1417–1427, Aug. (1975).

Design and Implementation of a Random Data-Placement System with High Scalability, Reliability and Performance

Kun Liu, Wei Xue, Di Wang, and Jiwu Shu

Dept. of Computer Science and Technology, Tsinghua University, Beijing
Liukun04@mails.tsinghua.edu.cn

Abstract. As storage system scales to thousands of disks, data distribution, load balance and the support for heterogeneous disks become increasingly important. In this paper, we present a new data-placement method named Weighted Interval Algorithm (WIA) for heterogeneous disks. Through it is not optimal in some circumstances, the difference between WIA and the optimal algorithm is trivial. Combined with replication, WIA can nearly balance access load and space utilization and improve reliability simultaneously. For the first time, we implement a data-placement system with high scalability, reliability and performance. The experimental results show that WIA reduces the average response time by 14.8% and decreases coefficient of relative load from 78.09% to 47.46% while the difference of the ratio of space utilization between disks is not more than 0.79%.

1 Introduction

With the development of computer technology, more and more information is created and used in a digital way. When this trend makes it easier for us to utilize information, it also makes the storage system's scalability, reliability and performance become increasingly important. This domain has become a focus and difficulty of research [1], [2], [3], [4], and [5].

Data placement is an effective way to improve scalability of storage system. It maps data objects to different disks to balance access load and space utilization. There are two well-known data-placement policies: striping and random data-placement [4]. Compared to striping, random data-placement moves almost optimal data objects to achieve new balance when the characteristic of storage system changes and so has better scalability. What's more, its performance is good and comparable to striping [3].

At the same time, data placement introduces the problem of reliability. It may place logically related objects to different disks and each disk can become single point of failure. And heterogeneous disks are inevitable in a massive storage system, so determining how to distribute data to heterogeneous disks is another problem faced by data placement.

At present, most researches on data placement are about the theoretic model and little attention is paid to the reliability of data-placement system. Furthermore, data placement based on disk capacity is hard to balance access load and space utilization at the same time. In this paper, in order to describe the power of disks, we define the

weight of disk, which can be the capacity, throughput or some combination of the two. Then a new data placement method named Weight Interval Algorithm (WIA for short) is proposed. Combined with replication, WIA can balance the access load and space utilization and improve reliability at the same time. For the first time, we implement the data-placement method in a real storage system AXUM, which is an in-band virtualization system and introduced in section 3, and get some valuable experimental data to support this method.

2 Related Work

Liner Hashing and its variants [9], [10], [11], and [12] adopt scalable distributed data structure. This method does not take into account the differences between disks and so does not support heterogeneous disks. An algorithm for pseudo-random distribution of data to multiple disks using partitioning of the unit range is proposed in [2]. But it doesn't allow for the placement of replicas and isn't optimal even in theory.

The algorithm in [1] can place data objects optimally. One flaw of this method is that it supports heterogeneous disks in a restricted way. The concept of cluster is introduced in this algorithm, which is a group of homogeneous disks. Disks are added into the system in a cluster way, which means only homogeneous disks can be added to the system at a time. And the data object and its replicas must be placed at the same cluster. This may introduce heavy load imbalance between clusters. Another flaw is that though heterogeneous disks can be added, the construction of a storage system must begin with a lot of homogeneous disks. The reason why cluster is introduced is that under some circumstances the optimal algorithm for replicas placement does not exist but with the prerequisite of cluster, the algorithm in [1] is optimal. On the contrary, our algorithm, WIA, has no prerequisites. Disks can be added into storage system in any way. And the difference between WIA and the optimal algorithm is rather trivial.

At present, data-placement algorithms are mostly aimed to balance the space utilization of disks; little attention is paid to access load. The method in [13] is to place data objects based on B-ZBSR, bandwidth-zone bandwidth to space ratio. Though it can alleviate the imbalance of access load, it wastes a lot of disk space.

Generally speaking, there are two policies to improve reliability: replication and parity. The latter is complicated and hard to scale. In [8], data object and its replica are placed to neighboring disks to improve reliability. This method can not utilize the parallelism of multiple disks adequately. If a disk has many hotspots, then most of the access is focused on two disks while others are free.

3 AXUM: An In-Band Virtualization System

AXUM is an in-band virtualization system and our platform for data placement. All physical disks (PD for short) are integrated into a storage pool named Source Container. Every PD is partitioned into several segments named Storage Granularity (SG for short) with the same size. According to the requirement, the administrator can allocate some storage space, that is to allocate some SG from SC to form a Virtual Disk (VD for short), which is used by users. So a VD is also composed of different SGs,

which are located in different PDs. Then the function of data placement on the basis of AXUM is to map each SG in VDs to a unique SG in different PDs to achieve a high scalability and performance.

4 Weighted Interval Algorithm

4.1 The Model and the Criterion

Assume that there are n physical disks $PD_1 PD_2 \dots PD_n$ in the storage system and the weight of PD_i is w_i . The weight can be the capacity, throughput or some combination of the two. Define the weighted interval for PD_1 as $(0, w_1)$, PD_2 as $(w_1, w_1 + w_2)$, PD_n as $(\sum_{i=1}^{n-1} w_i, \sum_{i=1}^n w_i)$. The data object set to be distributed is $S = \{SG_1, SG_2 \dots SG_m\}$ and m is the size of S .

We can evaluate different data placement algorithms according to the following criterions [2].

1. **Faithful distribution**, i.e. distributing a set of objects among a set of disks in such a way that the fraction of objects stored at a disk is equal (or at least close) to its share of the total weight of the system.
2. **Efficient localization**, i.e. computing the position of an object with a low time and space complexity.
3. **Fast adaptation**, i.e. adapting to changing weight with a near-minimal movement of objects.

4.2 Weighted Interval Algorithm

When replicas are not taken into account, WIA is similar to the algorithm in [1]; the difference is that we get rid of the restriction of cluster so it becomes much more flexible and simple. The basic idea of WIA is that for every data object SG_i in the set

of S a random number r is generated, which distributes evenly between 0 and $\sum_{i=1}^n w_i$.

The object SG_i is placed to PD_i when r belongs to the weighted interval of PD_i . Under this circumstance, this algorithm can distribute data objects faithfully [1].

4.3 The Placement of Replicas

The principle of the placement of replicas is that no two replicas of a data object can be placed at a same disk. No algorithm can satisfy both the criterion in 3.1 and the principle above because under some circumstance replicas of a SG have to be placed at a same disk if faithful distribution is satisfied firstly.

So an approximation algorithm is proposed. Assume $PD = \{PD_1, PD_2 \dots PD_n\}$; the data set to be placed is $S = \{SG_1, SG_2 \dots SG_m\}$ and each SG_i has a certain number of replicas. For each SG_i , SG_i is placed by WIA at PD_1 for example. Then the first replica of SG_i is distributed to $PD - \{PD_1\}$ by WIA, assume the result is PD_2 . The rest replicas are distributed to $PD - \{PD_1, PD_2\}$ in the same way.

Assume the number of data objects is N and each has a replica. Then the number of data objects placed at PD_i is $S_{oi} = N \frac{w_i}{w}$; the number of replicas placed at PD_i

is $S_{ci} = \sum_{j \neq i} N \frac{w_j}{w} * \frac{w_i}{w - w_j} = N \frac{w_i}{w} \sum_{j \neq i} \frac{w_j}{w - w_j}$. So the total number of data objects

placed at PD_i is $S_i = S_{oi} + S_{ci} = N \frac{w_i}{w} (1 + \sum_{j \neq i} \frac{w_j}{w - w_j})$.

Let $w_0 = \sum_{i=1}^n \frac{w_i}{w - w_i}$, then $S_i = N \frac{w_i}{w} (1 + w_0 - \frac{w_i}{w - w_i})$. This means this method trends to place data in disks with small capacity. The ratio of space utilization of PD_i is $\frac{S_i}{w_i} = \frac{N}{w} (1 + w_0 - \frac{w_i}{w - w_i})$ and the difference of ratio between disks is

decided by w_i . In a massive storage system, w_0 and w are much bigger than w_i . So the difference is trivial.

5 Implementation of the Data-Placement System

5.1 Weight, Hotspot Replication and Load Balance

The weight in WIA is a significant parameter to determine the number of data objects placed at a disk. Generally speaking, two main parameters depicting a disk are the capacity and throughput. Throughput is not a good choice of weight. Firstly, the throughput of disk is determined by the disk interface, average seeks time, rotational speed and access pattern so it is not a constant. Secondly, the development of disk throughput is much slower than that of disk capacity and if objects are placed according to throughput, a lot of disk space will waste. Finally, under practical environment hotspot does exist so the weight of throughput can not ensure maximal bandwidth. The space utilization of each disk is the same if the weight is disk capacity. But because of the existence of hotspot, the load of each disk will be unbalanced.

Based on our analysis of the HP trace [6] such as cello 92, we observe that disk access has a high degree of space locality and very often less than 20% data objects serve more than 90% access. So, if replicas of hotspot, named hotspot replicas, are created and placed on the disk with light relative load, then this can not only improve

performance but also balance access load. The relative load is defined as the ratio of disk access frequency to disk bandwidth.

We update the replicas and the original data object synchronously to ensure the consistency of data. The number of hotspot replicas has a great influence. If the replicas number is too small, access load of disks can balance to some degree but there is still some space to go on. If the replicas number is too large, though the access load balances well, because every write should update many disks synchronously, the additional load can outweigh the advantage of load balance and the total performance of the system would degrade. So we calculate the number of hotspot replicas in the following way. Record the access frequency of each SG and calculate the average access frequency (AAF for short). Define a parameter *threshold*, which is an adjustable number. If the access frequency of a SG is more than the product of average access frequency and the threshold, then this SG is a hotspot and replicas should be created. The number of replicas is determined by $\left\lceil \log \frac{\text{access_frequency}}{AAF * \text{threshold}} \right\rceil + 1$. The reason why logarithm is used is that logarithm can ensure that the number of hotspot replicas is moderate.

5.2 Reliable Replicas and Overhead

Besides balancing load, replicas can improve reliability [7], [8]. In an ideal data placement system, the crash of a physical disk can degrade the service of the whole storage system dramatically. Then replicas, named reliable replicas, can be created for each SG to improve reliability. These replicas can be placed by the algorithm in 4.3, and if C replicas are created for SG, the system can still work well when no more than C disks fail. The number of reliable replicas for a SG is determined by the importance of SG and how to get the importance is out of the scope of this paper.

Replication introduces two kinds of overhead. Firstly, excess space is needed to store replicas. In practice, the data stored at disks are much more valuable than the medium storing them. And with the fast increase of disk capacity and rapid decrease of disk price, we consider the additional disk space is worthy and an example is Google File System [14]. The second overhead is caused by synchronous update for writing replicas, which is unavoidable but can be alleviated. Firstly, not all hotspot are suitable for replication. When the majority of access to a hotspot is writing, then replication of that hotspot will degrade performance. Secondly, since hotspot replicas can also act as reliable replicas, if hotspot already has reliable replicas, then these reliable replicas can be changed to hotspot replicas.

6 Experiment Evaluation

6.1 Evaluating Data Placement

Figure 1 illustrates the ratio of data objects to be migrated when additional storage capacity is added to AXUM. Striping achieves scalability at a very high cost; almost every data object has to be moved. The number of data objects moved by WIA is very close to the

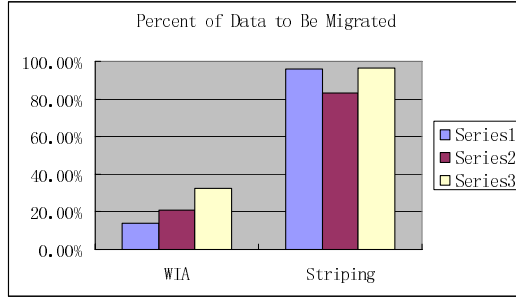


Fig. 1. Percent of data to be migrated when additional storage capacity is added to AXUM. Series 1 shows that 13.42% of storage capacity is added; series 2 shows 20.13% is added and series 3 shows 33.55% is added.

optimal ratio. Our experiment also shows that WIA performs better for random I/O while striping is better for sequential I/O. This is consistent with the results in [4].

6.2 Evaluating Replication

If we define coefficient of relative load as the ratio of the difference between maximal relative load and minimal relative load to maximal relative load, then that value for AXUM without hotspot replicas is 78.09%, while the optimal value is 0. This shows the necessity of hotspot replicas.

Figure 2 is the result after hotspot replicas are created. The horizontal line in the figure is the average response time when there are no hotspot replicas. The result is the same as the discussion in section 5.1.

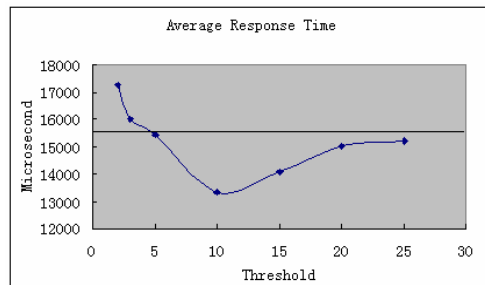


Fig. 2. Threshold and the corresponding average response time. The unit of time is microsecond. This experiment makes it clear that threshold has a great influence on the performance, and 10 is a good choice.

The coefficient of relative load is 47.46% after hotspot replicas are created with the threshold of 10. Of course we can create more hotspot replicas by decreasing threshold to make the coefficient of relative load drop to 0, but this will increase overhead of synchronous update and degrade the performance.

As to reliable replicas, in our experiment, we create 2 reliable replicas for each SG and pull out one and two disks from disk array and test the function and measure the average response time of AXUM. The result is that every request gets a perfect response which means the integrity of data objects is guaranteed and the average response time increases by 12.5% when a disk fails.

7 Conclusion and Future Work

This paper proposes Weight Interval Algorithm for data placement, which is simple, flexible and can be applied to any case. Through it is not optimal in theory, the experiment data shows that in practice it can obtain a satisfying result. Based on this algorithm, we build a data-placement system. And to balance disk access and improve reliability, reliable replicas and hotspot replicas are introduced into this system. By experiment, we prove that this data-placement system is of high scalability, reliability and performance.

Data placement is about the issue of how to distribute data objects to different disks to improve performance and scalability. And the issue of data layout, which is about how to place data objects in a disk to reduce seek time and rotational delay, is one direction for further research. On the other hand, data placement can be easily extended to information lifecycle management, which places data not only according to the characteristic of storage medium but also the importance of data.

References

1. R. J. Honicky and E. L. Miller. A fast algorithm for online placement and reorganization of replicated data. In Proceedings of the 17th International Parallel & Distributed Processing Symposium, Nice, France, Apr. 2003.
2. Andre Brinkmann, Kay Salzwedel, and Christian Scheideler. Compact, Adaptive Placement Schemes for Non-Uniform Capacities. In Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA), pages 53-62, Winnipeg, Manitoba, Canada, Aug. 2002.
3. Beomjoo Seo. Survey on Data Placement and Migration Algorithms in Distributed Disk Systems, In Proceedings of 2004 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04), Las Vegas, Nevada, USA, June 21-24, 2004.
4. Jose, Richard, Berthier. Comparing Random Data Allocation and Data Striping in Multimedia Servers. In Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, Santa Clara, California, United States, Pages: 44 - 55.
5. Roger Zimmermann Shahram Ghandeharizadeh. Continuous Display Using Heterogeneous Disk-Subsystems. In Proceedings of the fifth ACM international conference on Multimedia Seattle, Washington, United States. Pages: 227 - 238.
6. Chris Ruemmler and John Wilkes. UNIX disk access patterns. In Proceedings of the Winter'93 USENIX Conference, January 1993, Pages 405-420.

7. John Wilkes, Richard Golding, Carl Staelin, and Tim Sullivan. The HP AutoRAID hierarchical storage system In ACM Transactions on Computer Systems, Pages: 14 (1):108-136.
8. Akitsugu WATANABE Haruo YOKOTA. Adaptive Overlapped Declustering: A Highly Available Data-Placement Method Balancing Access Load and Space Utilization 21st International Conference on Data Engineering (ICDE 2005).
9. W. Litwin, J. Menon, and T. Risch. LH* schemes with scalable availability. Technical Report RJ 10121 (91937), IBM Research, Almaden Center, May 1998.
10. W. Litwin, M. Neimat, G. Levy, S. Ndiaye, T. Seek and T. Schwarz. LH*s: a high-availability and high-security scalable distributed data structure. In proceeding of the 7th International Workshop on Research Issue in Data Engineering. 1997, Birmingham, UK, Apr. 1997. IEEE, Pages 141-150.
11. W. Litwin and M.-A Neimat. High-availability LH* schemes with mirroring. In proceeding of the Conference on Cooperative Information Systems, 1996, Pages 196-205.
12. W. Litwin, M-A. Neimat, and D. A. Schneider. LH*-a scalable, distributed data structure. ACM Transactions on Database Systems, 1996, Pages: 21(4):480-525.
13. Yong-Sook Park, Jeong-Won Kim, Ki-Dong Chung. A Continuous Media Placement using B-ZBSR on Heterogeneous MZR Disk Array In Proceedings of International Workshops on Parallel Processing, 1999,Pages: 482-487.
14. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System, In Proceedings of the nineteenth ACM symposium on Operating systems principles Bolton Landing, NY, USA, Pages: 29 – 43.

Learning in a Multi-agent System as a Mean for Effective Resource Management

Bartłomiej Śnieżyński and Jarosław Koźlak

AGH University of Science and Technology, Institute of Computer Science
Kraków, Poland
{sniezyn, kozlak}@agh.edu.pl

Abstract. In this paper symbolic, supervised learning is used in a multi-agent system for resource management. Environment is a Fish Bank game, where agents manage fishing companies. Rule induction is applied to generate ship allocation and cooperation rules. In this article system architecture and learning process are described and experimental results comparing performance of several types of agents are presented. The results obtained confirm that applying a supervised learning algorithm in a multi-agent system may improve resource management.

1 Introduction

One of the main problems in a modern society is a problem of efficient resource management. It may concern different domains like logistics, ecology, production planning or computer network designing. The management is especially difficult for common, renewable resources, i.e. the ones whose quantity in the environment increases until reaching a given maximum value, and the speed of this increase depends on the quantity of resources.

An exhaustion of common resources can be considered as an example of a crisis situation, which makes it impossible to regenerate them in the future and, as a consequence, makes the functioning of entities, which need these resources, impossible. Hence, it is important to guarantee a resource consumption on the appropriate level, and to minimize the probability of crisis situations arising as well as their scale.

The goal of our work is to provide a strategy of resource management and to guarantee that a system will stay in a safe state, i.e. the one, which does not lead to resources exhaustion.

To analyze such situations it is useful to apply multi-agent approach, which allows for modelling of activities of autonomous, rational entities – agents, which cooperate or compete one with another. Agents try to realize their goals in a way that is as good as possible, whereas a cooperation may be necessary to avoid an exhaustion of resources and appearing a crisis situation. The cooperation may occur as an auto-modification of actions concerning resources consumption performed by an agent or by negotiations and contracts.

One of the main problems in the development of such systems is designing an appropriate strategy for resource allocation and determination, when negotiations and contracts should be introduced and which character they should have. Applying learning algorithms allows to overcome such problems. One can implement an agent that is not perfect, but it improves its performance.

In this paper results of application of symbolic, supervised learning in a multi-agent system are presented. As an environment Fish Bank game is used [1]. It is a simulation where agents run fishing companies that must decide how much, and where to fish.

In the following sections learning in the multi-agent systems is described, developed system and a learning process used is presented, and experimental results are analyzed.

2 Learning in Multi-agent Systems

The problem of learning in multi-agent systems may be considered as a union of research on multi-agent systems and on machine learning. Machine learning focuses mostly on research on isolated process performed by one intelligent module. The multi-agent approach concerns the systems composed of autonomous elements, called agents, whose actions lead to the realization of given goals. In this context, learning is based on the observation of the influences of activities, performed to achieve the goal by an agent itself or by other agents. Learning may proceed in a traditional – centralized (one learning agent) or decentralized manner. In the second case more than one agent is engaged in the learning process [2].

The learning process is strictly associated with such aspects of agents as reasoning and decision making. The most popular learning technique in multi-agent systems is reinforcement learning that allows to learn what action should be executed in a current situation. Other techniques can be also applied. Learning process can be based on the symbolic knowledge representation (rules, decision trees), neural networks, models coming from game theory as well as optimization techniques (like the evolutionary approach, tabu search etc.).

An important feature of cooperative learning is the necessity of the interactions between agents using a suitable protocol. Thanks to these interactions, an agent may acquire additional knowledge useful to its decision making and allowing it to make a common decision in accordance with the preferences of the agents. An example may be a solution presented in [3]. Here, agents may come forward with proposals, which are expressed by an additional parameter which describes the confidence of the agent about them. In their response, the other agents participating in the negotiation process may accept the proposal (confirm), reject it (disagree), propose modifications (modify) or do not have an opinion on the subject (no opinion). In each case, except the last one, agents also declare their confidence concerning their evaluation.

3 System Description

Although Fish Banks game is designed for teaching people effective cooperation in using natural resources [4] it suits to using in multi-agent systems very well [1, 5]. In this research the game is a dynamic environment providing all necessary resources, action execution procedures, and time flow (game rounds). Each round consists of the following steps: ships and money update, ship auctions, trading session, ship orders, ship allocation, fishing, and fish number update.

Agents represent players that manage fishing companies. Each company aims at collecting maximum assets expressed by the amount of money deposited at a bank account and the number of ships. The company earn money by fishing at fish banks.

Environment provides two fishing areas: coastal and a deep-sea. Agents can also keep their ships at the port. Cost of fishing at the deep-sea is the highest. Cost of staying at port is the lowest but such ship does not catch fish.

Initially, it is assumed that the number of fish in both banks is close to the bank's maximal capacity. During the game the number of fish in every bank changes according to the following equation:

$$f_{t+1} = f_t + bf_t\left(1 - \frac{f_t}{f_{max}}\right) - C_t, \quad (1)$$

where f_t is a fish number at a time t , b is a birth rate (0.05 value was used in experiments), f_{max} is a maximum number of fish (equal 4000 for a deep sea, and 2000 for a coastal area), $C_t = nc_t$ is a total fish catch: n is a number of ships of all players sent to the bank, and c_t is a fish catch for one ship at the time t :

$$c_t = c_{max}w_t\sqrt{\frac{f_t}{f_{max}}}, \quad (2)$$

where c_{max} is a maximal catch (equal 25 for a deep sea, and 15 for a coastal area), and w_t is a weather factor at a time t , which is a random number between 0.8 and 1.0. As we can see, at the beginning of game, when f_t is close to f_{max} , fishing at the deep sea is more profitable.

Usually exploration overcomes birth and after several rounds the fish number can decrease to zero. It is a standard case of "the tragedy of commons" [6]. It is more reasonable to keep ships at the harbor then, therefore companies should change their strategies.

Three types of agents are implemented: learning agent, predicting agent, and random agent. The first one uses experience to allocate ships, the second one uses previous fishing results to estimate values of different allocation strategies, third one allocates ships randomly. All types of agents may observe the following aspects of the environment: new ships that they receive from a shipyard, money earned in the last round, ships allocations of all agents, and fishing results (c_t) for a deep sea and an inshore area. All types of agents can execute the following two types of actions: order ships, allocate ships.

Order ships action is currently very simple. It is implemented in all types of agents in the same way. At the beginning of the game every agent has 10 ships. Every round, if it has less than 15 ships, there is 50% chance that it orders two new ships.

Ships allocation is based on the method used in [1]. Allocation strategy is represented by a triple (h, d, c) , where h is the number of ships left in a harbor, d and c are numbers of ships sent to a deep sea, and a coastal area respectively. Both types of agents generate a list of allocation strategies for $h = 0\%, 25\%, 50\%, 75\%$, and 100% of ships that belong to the agent. The rest of ships (r) is partitioned; for every h the following strategies are generated:

1. All: $(h, 0, r), (h, r, 0)$ – send all remaining ships to a deep sea or coastal area,
2. Check: $(h, 1, r - 1), (h, r - 1, 1)$ – send one ship to a deep sea or coastal area and the rest to the other,
3. Three random strategies: $(h, x, r - x)$, where $1 \leq x < r$ is a random number – allocate remaining ships in a random way,
4. Equal: $(h, r/2, r/2)$ – send equal number of ships to both areas.

The random agent allocates ships using one of strategies chosen by random. Learning agent does the same in the first game, but in the following games it chooses strategy with the highest rating. Strategy rating is generated using rules that allow to classify allocation as *good* or *bad* taking into account allocation parameters and environment parameters (fish catch at the deep sea and at the coastal area in the previous round).

Every strategy s gets a value v according to the formula:

$$v(s) = \alpha \text{good}(s) + \text{bad}(s), \quad (3)$$

where $\text{good}(s)$ and $\text{bad}(s)$ are numbers of rules, which match the strategy and current environment parameters, with consequence *good* and *bad*, respectively, and α is a weight representing importance of rules with consequence *good*. The rules are learned using agent experience (see section 4).

If there is more than one strategy with the same value, one occurring earlier in the list is chosen.

Predicting agent uses the following formula to estimate value of the strategy:

$$v(s) = \text{income}(s) + \eta \text{ecology}(s), \quad (4)$$

where $\text{income}(s)$ represents prediction of the income based on the previous fishing results, $\text{ecology}(s)$ represents ecological effects of the action s (the value is low if fishing is performed in the area with low fish population), and η represents importance of the ecology factor.

Software used in experiments is written in Prolog, using Prologix compiler [7]. Every agent is a separate process. It can be executed on a separate machine. Agents communicate with the environment using Linda blackboard.

Table 1. Predicates used to describe the event

Predicate	Argument values	Description
rate(R)	good, bad	Rating of the allocation strategy
harbor(N)	100%, 75%, 50%, 25%, 0%	Fraction of ships left in a harbor
alloc(A)	100%-0%, 75%-25%, 50%-50%, 25%-75%, 0%-100%	Allocation: ship fraction sent to deep-ship fraction sent to coastal
prevCatchDeep(D)	integer numbers	Number of fish caught by every ship on a deep sea
prevCatchCoastal(C)	integer numbers	Number of fish caught by every ship on a coastal area

4 Learning

To support learning AQ21 program is used [8]. It is a machine learning software that allows to generate attributional calculus rules for given examples. In a standard mode it produces a complete and consistent description of the data, but it can also provide rules that are not complete and/or consistent. The main advantage of this program is that generated rules are not ordered and the knowledge is easy to interpret for human what makes experimental results easier to check and can be useful in Fish Bank application to teach people. Of course, other methods of classifier learning can be used.

4.1 Ship Allocation Learning

The AQ21 program generates a classifier that is used to rate ship allocation strategies. Input attributes for the classifier are allocation parameters (ratios of ships that are kept in a harbor, sent to the deep sea, and sent to a coastal area) and environment parameters. Target attribute is a rating of the allocation in a given environment state. It has two values: good and bad. Predicates used to describe the event are presented in Tab. 1.

Training events are generated from agents observations. Every round the learning agent stores ship allocations of all agents, and fish catch in the previous round. The strategy of an agent with the highest income is classified as good, and the strategy of an agent with the lowest income is classified as bad. If in some round all agents get the same income, none strategy is classified, and as a consequence, none of them is used in learning.

At the end of each game the learning agent uses events, which were generated during all games played so far, to learn a new classifier that is used in the next game.

4.2 Cooperative Learning

The next step in model development is to introduce cooperative learning of agents to avoid depleting of fish. There are three main aspects of negotiations that are considered:

- selection of the time point when negotiation is performed. If an agent discovers that its situation is worse than in a previous tour, it starts a negotiation;
- selection of a proposal to be negotiated. The selection may be performed by random with preference to decision which proved to be the most advantageous in the most of situations or for the most situations in similar cases;
- decision on whether the negotiated proposition should be accepted or rejected. Agents evaluate a list of proposals (each agent accepts or rejects them) and then select the one with the highest degree of acceptance. If an agent does not have any reason to accept or reject negotiation results (i.e. does not know if the consequences of a particular condition is good or bad for it) then it makes a decision by lot. Proposal has to get a majority of the agents to be valid. At the end of game an agent assesses whether the accepted proposition improved or worsened its situation and classifies the proposal as good or bad for the given system state.

To represent the agent situation, the wealth of the agent, numbers of fish on the banks, and/or trends concerning the change of wealth or the change of the number of fish caught should be taken into account.

These attributes are used to generate rules that are used to rate proposals. Learning rules after a game can improve negotiation strategies. As for ship allocation learning, AQ21 learning program can be applied.

5 Experimental Results

To test how learning changes an agent performance four series of experiments were performed. Four agents took part in every series. Every series consisted of ten simulations, and each simulation consisted of the sequence of 15 games.

In the first series there were three random agents and one learning agent (with weight $\alpha = 1$). Performance of the agents measured as a balance at the end of every game is presented in Fig. 1-(a).

In the second series one learning ($\alpha = 1$), one predicting and two random agents were used. Performance of agents is presented in Fig. 1-(b).

In the third and fourth series there were two learning and two random agents. In both series, the first learning agent (LA1) has $\alpha = 1$. In the former series, presented in Fig. 1-(c), second learning agent (LA2) has $\alpha = 2$, in the latter, presented in Fig. 1-(d), LA2 has $\alpha = 3$.

In all experiments average balance of the learning agents grows with the agent's experience, while performance of the predicting agent decreases slightly (because of the learning agent competition). Performance of random agents generally doesn't change.

Examples of rules learned are presented in Fig. 2. Capital letters represent variables that can be unified with any value. Predicate `member` checks if its first argument belongs to the list that is a second argument. It is used to represent an internal disjunction (expression of the form $x = v_1 \vee v_2 \vee \dots \vee v_n$). Remaining predicates are explained in the previous section. These rules (in the form of clauses) can be interpreted in the following way.

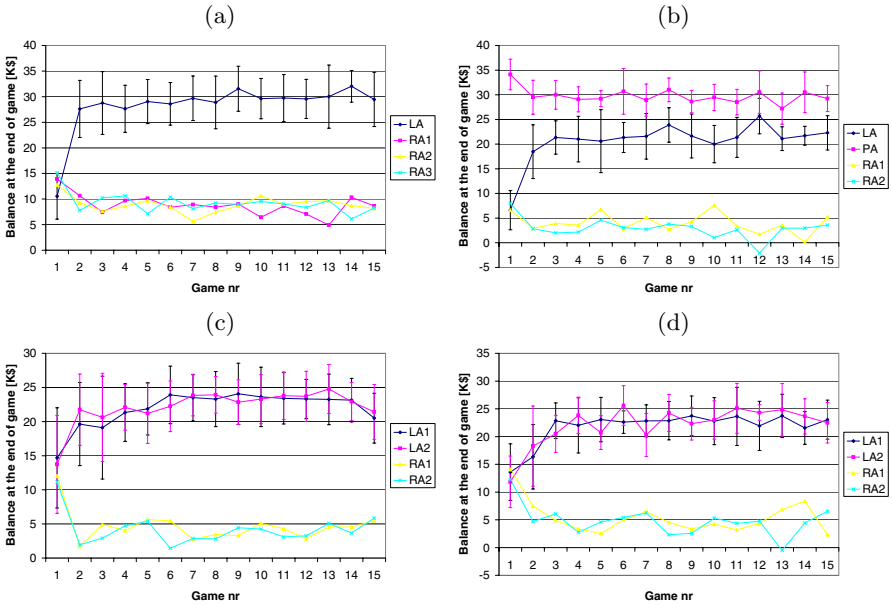


Fig. 1. Comparison of performance of learning agents (LA, LA1, LA2) and other agents using random strategy of ship allocation (RA1, RA2, RA3) or prediction (PA); values for learning and predicting agents are presented with the standard deviation

```

(a)
rate(bad) :-
    harbor(B),
    member(B, [25,50,75]),
    prevCatchDeep(C),
    C >= 16,
    prevCatchCoastal(10).

(b)
rate(good) :-
    alloc(B),
    member(B, [100%-0%,75%-25%]),
    prevCatchDeep(C),
    C >= 18,
    C <= 21,
    prevCatchCoastal(D),
    D <= 10.
    
```

Fig. 2. Examples of rules (in the form of Prolog clauses) learned by the agent

Clause (a): it is a bad decision to keep at a harbor 25, 50, or 75 percent of ships if previous catch at a deep sea is greater or equal to 16, and previous catch at a coastal area is 10.

Clause (b): it is a good decision to send 100% ships to a deep sea or 75% to a deep sea and 25% to a coastal area if previous catch at a deep sea is greater or equal to 18, and smaller or equal to 21, and previous catch at a coastal area is smaller or equal to 10.

Experimental results show that the learning agent performance increases rapidly at the beginning of the learning process, when generated rules are used instead of a random choice. Next it increases slowly, because new examples do not contain any significant knowledge. The performance stabilizes at the end of the process.

Weight α has no significant influence on the learning agent performance. However values greater than 1 seem to be a little bit better than $\alpha = 1$.

As we can see in Fig. 1-(b), the predicting agent performs better than the learning agent. It suggests, that there is a space for improvement of the learning method. Further research is necessary to check if it is possible to learn such a good strategy.

6 Conclusion and Further Research

The results obtained confirm that applying a symbolic machine learning algorithm in a multi-agent system may improve resource management.

Fish Banks environment is complex enough to apply supervised learning, when direct performance feedback is available (e.g. income at the end of the round), and, if there is no such information, and feedback is available at the end of game (reinforcement learning can be used in such situation). Testing the latter case is a subject of further research.

Acknowledgments. The authors thank prof. Ryszard S. Michalski, George Mason University for providing AQ21 program, and Arun Majumdar, Vivomind Intelligence Inc. for providing Prologix system, which was used in the implementation.

References

1. Kozlak, J., Demazeau, Y., Bousquet, F.: Multi-agent system to model the fishbanks game process. In: The First International Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAS'99), St. Petersburg (1999)
2. Sen, S., Weiss, G.: Learning in multiagent systems. In Weiss, G., ed.: A Modern Approach to Distributed Artificial Intelligence. The MIT Press (1999)
3. Sian, S.: Adaptation based on cooperative learning in multi-agent systems. In Demazeau, Y., Muller, J.P., eds.: Decentralised AI (Vol. 2). (1991) 257–272
4. Meadows, D., Iddman, T., Shannon, D.: Fish Banks, LTD: Game Administrator's Manual. Laboratory of Interactive Learning, University of New Hampshire, Durham, USA (1993)
5. Sniezynski, B., Kozlak, J.: Learning in a multi-agent approach to a fish bank game. In: Multi-Agent Systems and Applications IV: Proc. of CEEMAS 2005. Volume 3690 of Lecture Notes in Computer Science. (2005) 568–571
6. Hardin, G.: The tragedy of commons. *Science* **162** (1968) 1243–1248
7. A. Majumdar, P. Tarau, J.S.: Prologix: Users guide. Technical report, VivoMind LLC (2004)
8. Wojtusiak, J.: AQ21 User's Guide. Reports of the Machine Learning and Inference Laboratory, MLI 04-3. George Mason University, Fairfax, VA (2004)

Multicriterial Decision-Making in Multiagent Systems

Petr Tučník¹, Jan Kožaný², and Vilém Srovnal¹

¹ Department of Measurement and Control
VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba
Czech Republic
petr.tucnik@vsb.cz, vilem.srovnal@vsb.cz

² Department of Computer Science, FEECS,
VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba
Czech Republic
jan.kozany@vsb.cz

Abstract. The main purpose of this article is to present multi-criteria decision-making principle as a tool providing the agent with autonomous decision-making capacity. The advantages and disadvantages of this principle are described on the example of the robot soccer game, together with the future perspective of this approach. The control system for robot soccer game is designed with consideration of using a multi-criteria decision-making. As a quickly changing environment, the robot soccer game provides an excellent testing ground for such experimental approach.

1 Introduction

The multi-criteria decision-making (MDM) problem area is laying in the intersection of several branches of science. Today, it is mainly used in the decision-support systems. From the point of view of the artificial intelligence the possibility of application of the MDM principle for governing the agent actions autonomously is more interesting. It allows us to implement various methods of machine learning and leads us to unaccustomed perspective of viewing machine decision-making. Therefore, this text is focusing the autonomous decision-making and the area of decision-support will be generally omitted.

2 Basic Terms

Before we try to deal with the explanation of the MDM principle itself, we will define the basic notions. Then we will use these defined notions in the means of following descriptions, definitions and equations.

The agent is the set $X = \{A, S, M\}$, where $A = \{A_1, \dots, A_n\}$ is a nonempty set of actions realized by the actuators, $S = \{s_1, \dots, s_m\}$ is the nonempty set

of *scanning functions* and $M = \{w_1, \dots, w_k\}$ is the nonempty set of motivational functions and coefficients that characterizes the goals of the agent and the effectiveness of its actions (through the motivational functions it is possible to implement machine learning methods).

The actions of the set A are divided into the subsets accordingly to the actuator that is capable of their realization. Motivation value of the specific actuator selection is divided between actions that this actuator is capable of. With each actuator at least one specific motivational function is assembled, when there is a large scalability of the possible actuator actions, more motivational functions may be presented. As a supportive tool we may use the initiatory conditions or threshold values. The whole functional spectrum of the actuator activity (its whole extent of activities) has to be covered.

The notion of *attribute* denotes one (distinguishable) characteristic feature of the agent's environment and it is represented numerically. This is important condition for the successful application of the MDM method.

The *universum* is the environment described from the agent's point of view as a system of sets U , resp. z -nary vector of numbers (attributes), where $z \in N$. The z -nary vector represents all the attributes in the universum. For the universum, the following equation is valid:

$$U = U_R \cup U_I \cup X, \quad (1)$$

where U_R represents real component of the world, U_I imaginary component of the world and X agent itself. The universum may not be considered valid without the presence of the agent itself, as it is an inseparable part of it. The imaginary part of the universe may not be omitted either, as it may influence the decision making process of the agent. The universum is therefore the sum of all influences relevant for the agent during the decision forming process.

The *universum-configuration* represents z -nary vector of attribute values in the particular time. The *attribute value* represents measure of presence of given characteristic in the environment. The attribute value fits into bordered interval the endpoints (*upper limit* and *lower limit*) of which are specified by the sensor sensitivity and range.

The *decision-making situation* represents such universum-configuration that it is necessary to start the *decision-making* process i.e. the sequence of steps leading to the selection of the best solution variant of the current situation.

With the notion of the variant we refer to the sequence of one or more actions from the set A (application of such actions must be possible in the given situation), which is in fact the transition from the starting state of the agent to the state that corresponds with the motivations of the agent in its decision-making moment.

In the universum, the area of the agent's effect P , is defined:

$$P = U \cap (A'_1 \cup \dots \cup A'_n), \quad (2)$$

where A'_i ($i = 0, \dots, n$) is the area of the effect of the actuator A_i .

The scanning function $s_j(a_1, \dots, a_p)$, in the universum U defined by the attributes (a_1, \dots, a_z) , returns scanned values of p -nary vector of attributes

(a_1, \dots, a_p) . This vector is scanned by the item s_j of the sensor set S . It holds up for $z \geq p$. For the function s_j , the following equations are valid:

$$s_j(a_1), s_j(a_2), \dots, s_j(a_p) = s_j(a_1, \dots, a_p), \tag{3}$$

$$\forall s_j \in S : R_e(s_j) \in \langle k, l \rangle \mid -\infty \leq k \leq l \leq \infty, j = 1, \dots, m, \tag{4}$$

$$\forall s_j \in S : Im(s_j) \in \langle 0, 1 \rangle, j = 1, \dots, m, \tag{5}$$

while during the signal processing, the standard normalization formula

$$\text{normalized_value} = (\text{attribute_value} - k) / (l - k) \tag{6}$$

is used. The attribute value is normalized to the interval of $\langle 0; 1 \rangle$ and variables k, l are expressing the lower and upper limit of the sensor scanning range. Concurrently, the following condition has to be fulfilled:

$$s_1(A'_1, \dots, A'_z) \cup s_2(A'_1, \dots, A'_z) \cup \dots \cup s_m(A'_1, \dots, A'_z) \supseteq P. \tag{7}$$

The agent has to be able to perform sensor scanning at least in the area of effect of its actuators. If the condition (7) is not fulfilled, the agent is not able to perform his activities effectively, as it is not able to perceive all changes made in the environment by its actuators.

3 The Basic Principle

In [5], [1], the wide scale of methods of multi-criteria decision-making may be found. The Fig. 1 shows the steps of the MDM process.

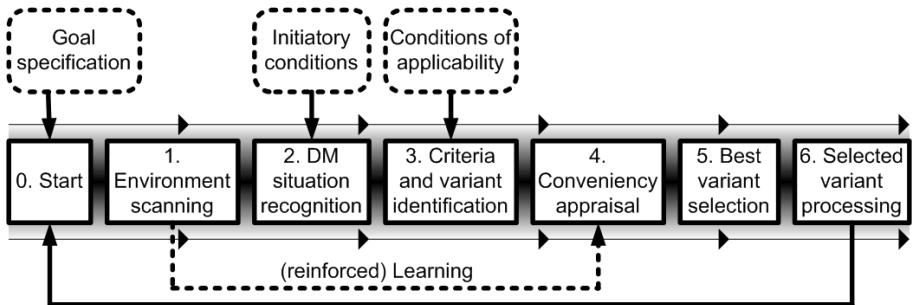


Fig. 1. The MDM process

After initialization and goal acquisition in the phase 0, the agent tries to refresh its environmental data by its sensors – this is the phase 1. The set of initiatory conditions is used for the decision-making situation recognition in the phase 2. It has to provide the agent with the ability to react to unexpected changes in the environment during its target pursue. Also, it must allow the agent to

change its goal through the decision-making process pertinently, e.g. if the prior goal is accomplished or unattainable. In the phase 3, the conditions of applicability characterize the boundary that expresses the relevance of using the variant in the decision-making process. The most important step is the phase 4, where the convenience appraisal of applicable variants is performed. This step plays the key part in the decision-making procedure. The following formula is used:

$$\text{convenience} = \sum_i w(\text{inv}(\text{norm}(a_i^v))), \quad i = 1, \dots, z, \quad (8)$$

where v stands for the total number of variants.

The convenience value for the each of assorted variants is obtained. Attributes a_i^v stand for presumptive values of universum-configuration and the *norm* function normalizes the value of the attribute and is mentioned above (6). The function *inv* is important, as it represents reversed value of difference between real attribute value and ideal attribute value:

$$\text{inv}(\text{current_value}) = 1 - (\text{ideal_value} - \text{current_value}). \quad (9)$$

The optimal variant remains constantly defined by m -nary vector of attributes, where $m \leq z$, for the each of the decision-making situations and attributes a_i^v differ for each variant other than the optimal variant. There is a final number of activities that the agent is able to perform. As the inverse values of difference between the real and ideal variant are used, in the most ideal case, the convenience value will be equal to 1, and in the worst case it will be close to 0. $\text{Im}(\text{inv}) = (0; 1)$. The lower open boundary of the interval is useful, because troubles related to computation with zero (dividing operations) may be avoided.

The function w assigns the importance value (weight) to the each attribute. The machine learning is realized by proper modifications of the weight function. Importance of attributes differs in accordance with the actual state of the agent. E.g. energetically economical solution would be preferred when the battery is low, fast solution is preferred when there is a little time left, etc. Precise definitions of weight functions are presented in [5], [1]. In the phase 5, the variant with the highest convenience value is selected and its realization is carried out in the phase 6. During processing of the selected solution, the agent is scanning the environment and if the decision-making situation is recognized, the whole sequence is repeated. The evaluation function (e.g. reinforced learning functions examples in [3], [4], [6]) provides the feedback and supports the best variant selection, as it helps the agent inits goal pursue. Based on the scanned environmental data, modifications of the function w are made during the learning process. The agent is optimizing selection process and gets abetter appreciation of the goal pursue issues. Function w belongs to the M set (motivational factors of the agent).

4 Attribute Composition and Decomposition

The attributes are a cornerstone of the MDM system. Therefore, their formulation requires presence of the proper tool. It is necessary to provide the MDM

system with the adequate amount of attributes. This is where the composition and decomposition technique is used. Properly functioning MDM system has to fulfill following condition:

$$|A| \geq 1 \wedge |S| \geq 1 \wedge |M| \geq 1, \quad (10)$$

which expresses the requirement of non-emptiness of the A , S , M sets of the agent. But the case:

$$|A| = 1 \wedge |S| = 1, \quad (11)$$

is trivial one. As there is only one attribute to assess and only one solution possible, it is not really a MDM system. Therefore, it is better to fulfill the following condition:

$$|A| \geq 2 \wedge |S| \geq 2 \wedge |M| \geq 1, \quad (12)$$

where two attributes are considered and two options are possible. There is only one evaluating function, but it must be included for it is necessary to provide the agent with satisfactory feedback. It is not possible to call the system artificially intelligent if it has no tendency to rational behavior in its environment. Such rational aspect may be ensured by implementation of the motivational factor (goal) to the agent. Therefore, at least one item in the M set of the agent is required. Many repeats of composition or decomposition brings up disserviceable results or imbalances and dependencies in the universum structure.

5 Examples

The actual aim of our research is the strategy in a robot control system. The strategy is a component in a hybrid control architecture of the robot soccer control system [2]. This component has a recognized picture from camera (robots' and ball's coordinations) in its input. Then it have to make a correct real-time decision-making that depends on the current situation on the playground. The output of the strategy component is the set hardware level orders setting the wheel velocities on each our robot.

We have decided to apply the MDM method to the decision making problem. For successful application of the MDM method it is necessary to provide the sufficient amount of attributes to the decision-making system. The fact is that the only truly relevant data input are video camera pre-processed images. The decision-making system has to derive all the necessary data from this input, so we proposed the system of attributes that is based on the video recognition.

The main set of attributes consists of items such as position of the agent in the system of coordinates, its speed, heading angle, etc. The auxiliary data structure is division of the soccer playground into areas, as it is shown at Fig. 2. Such division provides additional (more detailed) information about the agent position and is useful for easier situation recognition. Hexagon representation in Fig. 2 is presented because it is easier to see the limited amount of transitions between the given positions (areas).

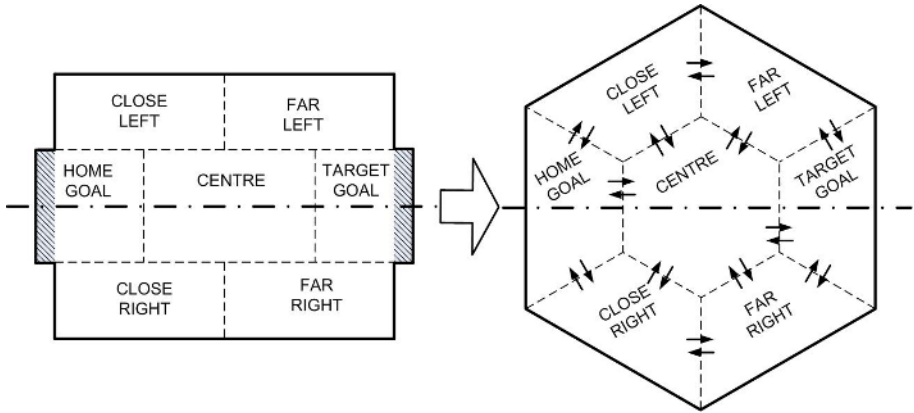


Fig. 2. Divided robot soccer playground and its hexagonal representation

For easier use of the MDM method it is useful to divide strategy component into several subcomponents in a hierarchical manner. We have adopted a hybrid control architecture of the robot soccer control system [2]. The hybrid control architecture with modified strategy component is shown at Fig. 3.

Preprocess component. This component preprocesses a recognized picture from camera – it predicts the ball's and robots' positions, all attributes needed for next MDM-based selection of strategies are computed there.

High-level component. Recognition of game situation on the playground and the application of the MDM-based selection of the team strategy is performed here.

Medium-level component. At this level the roles are delegated to each agent for example keeper, defender or attacker and there are delegated instructions to each robot for example moving, shooting etc.

Low-level component. Each agent has to compute its wheel velocities based on selected instruction in order to perform.

The cooperation of all layers of the strategy module is necessary to achieve satisfactory results. Such architecture also respects the requirements of multi-agent approach, as it is not a purely centralized system, but a tool for role delegation and work control. It is possible to apply the MDM principle to the most decision-making situations; therefore, it allows us to perform the tests of this method.

The high-level component recognizes the game situation on the playground. It recognizes whether our team should attack or defend. Attributes are ball's and robots' positions and ball possession plays an important role during the recognition process. Next step is the selection of actions. One action can be executed by more than one robot, for example the pass action. If our team

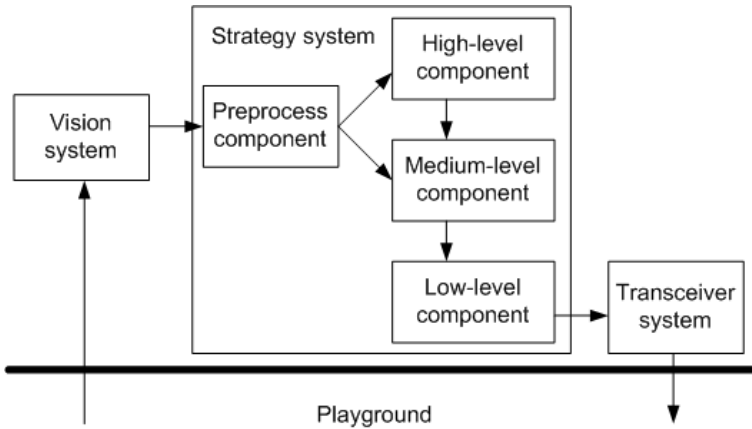


Fig. 3. Divided robot soccer playground and its hexagonal representation

attacks, the actions leading to score have the priority, and if our team is defending against opponent's actions then activities preventing opponent to score a goal have the priority. Attributes for a MDM based decision-making are based on the following facts - our team is in attack or not and on the positions of the robots and the ball. Evaluating functions also have to take the probability of the successful execution of actions under consideration.

The medium-level component is responsible for delegacy of roles and instructions. The roles during the action are delegated to the robots. If there are several robots assigned to one action then the robot with the most advantageous position for action execution obtains the role of a leader. Robots with the leader role are responsible for execution of the action. In the next step robots choose instructions leading to the successful execution of the action. The robots have to communicate with each other in order to choose instructions properly. The next reason for communication is need of the plan collision avoidance. Robots have to exchange their intentions and the adequate solution is the output of this communication process.

The low-level component is responsible for driving of our robots. There is no decision-making at this level. Robots don't communicate with each other.

6 Conclusion

The MDM principle represents an interesting approach to the complicated problem solving. However the main problem remainings in the proper virtual design of the environment. The universum of the agent must be represented in the numeric format and presented in a suitable quantity of attributes. Also, the motivations of the agent must be designed appropriately. In spite of these design problems, the MDM principle represents the versatile tool that is able to process the large variety of tasks.

Acknowledgment

The work and the contribution were supported by the project from Grant Agency of Czech Academy of Science - Strategic control of the systems with multiagents, No. 1ET101940418 (2004-2008).

References

1. Fiala, P., Jablonský, J., Manas, M.: Vícekriteriální rozhodování, VŠE Praha (1997).
2. Kim, J., Kim, D., Kim, Y., Seow, K.: Soccer Robotics (Springer Tracts in Advanced Robotics), Springer-Verlag, 2004
3. Kubík, A.: Agenty a multiagentové systémy, Silesian University, Opava (2000).
4. Pfeifer, R., Scheier, C.: Understanding Intelligence. The MIT Press, Cambridge, Massachusetts, (1999).
5. Ramík, J.: Vícekriteriální rozhodování – Analytický hierarchický proces (AHP), Silesian University, Opava (1999).
6. Weiss, G.: Multiagent Systems – A Modern Approach to Distributed Modern Approach to Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, (1999).

JADE-Based A-Team Environment

Piotr Jędrzejowicz and Izabela Wierzbowska

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{pj, iza}@am.gdynia.pl

Abstract. The paper proposes a JADE-based A-Team environment (JADE-A-Team) as a middleware supporting the construction of the dedicated A-Team architectures used for solving variety of computationally hard optimization problems. The paper includes a general overview of the functionality and structure of the proposed environment and a more detailed description of optimization agents including their standard functions, ontology, construction requirements and activation procedure. Further sections explain how to create and activate an A-Team agent and how the communication between agents is handled. Conclusions focus on advantages of the JADE-A-Team environment and on suggestions for further research.

1 Introduction

Recently, a number of agent-based approaches have been proposed to solve different types of optimization problems [1], [4], [5]. One of the successful approaches to agent-based optimization is the concept of A-Teams. An A-Team is composed of simple agents that demonstrate complex collective behavior.

The A-Team architecture was originally proposed by Talukdar [9] as a set of objects including multiple agents and memories which through interactions produce solutions of optimization problems. The advantage of the A-Team architecture is that it combines a population of solutions with domain specific algorithms and limited agent interaction. A sophisticated A-Team architecture was proposed in [7]. Some dedicated A-Teams were proposed in [6], [8]. According to [9] an A-Team is a problem solving architecture in which the agents are autonomous and co-operate by modifying one another's trial solutions.

In this paper we propose a JADE-based A-Team environment (in short: JADE-A-Team) as a middleware supporting the construction of the dedicated A-Team architectures used for solving variety of computationally hard optimization problems. JADE is an enabling technology, for the development and run-time execution of peer-to-peer applications which are based on the agents paradigm and which can seamless work and interoperate both in wired and wireless environment [2]. From the functional point of view, JADE provides the basic services necessary to distributed peer-to-peer applications in the fixed and mobile environment. JADE allows each agent to dynamically discover other agents and to communicate with them according to the peer-to-peer paradigm.

The paper contains a general overview of the functionality and structure of the JADE-A-Team and a more detailed description of optimization agents including their standard functions, ontology, construction requirements and activation procedure. Further sections explain how to create and activate A-Team agents. Final section explains how the communication between agents is handled. Conclusions focus on advantages of the proposed environment and on suggestions for further research.

2 Overview of the JADE-A-Team

The central problem in the design of a multi-agent system is how much intelligence to place in the system and at what level. As it was observed in [3], the vast majority of the work in this field has focused on making agents more knowledgeable and able. This has been achieved by giving the deliberative agent a deeper knowledge base and ability to reason about data, giving it the ability to plan actions, negotiate with other agents, or change its strategies in response to actions of other agents. At the opposite end of the spectrum lie agent-based systems that demonstrate complex group behavior, but whose individual elements are rather simple. The JADE-A-Team belongs to the latter class.

Its main functionality is searching for the optimum solution of a given problem instance through employing a variety of the solution improvement algorithms. The search involves a sequence of the following steps:

- Generating an initial population of solutions placing them in the common memory.
- Applying solution improvement algorithms which draw individuals from the common memory and store them back after an improvement, using some user defined replacement strategy.
- Continuing reading-improving-replacing cycle until a stopping criterion is met.

To perform the above two classes of agents are used. The first class includes *OptiAgents*, which are implementations of the improvement algorithms. The second class includes *SolutionManagers*, which are agents responsible for maintenance and updating of individuals in the common memory. All agents act in parallel. Each *OptiAgent* is representing a single improvement algorithm (simulated annealing, tabu search, genetic algorithm, local search heuristics etc.). An *OptiAgent* has two basic behaviors defined. The first is sending around messages on readiness for action including the required number of individuals (solutions). The second is activated upon receiving a message from some *SolutionManager* containing the problem instance description and the required number of individuals. This behavior involves improving fitness of individuals and resending the improved ones to the sender. A *SolutionManager* is brought to life for each problem instance. Its behavior involves sending individuals to *OptiAgents* and updating the common memory.

Main assumption behind the proposed solution is its independence from a problem definition and solution algorithms. Hence, main classes *Task* and *Solution* upon which agents act, have been defined at a rather general level. Interfaces of both classes include function *ontology()*, which returns JADE's ontology designed for classes *Task* and *Solution*, respectively. Ontology in JADE is a class enabling definition of the vocabulary and semantics for the content of message exchange between agents. More precisely, an ontology defines how the class is transformed into the text message exchanged between agents and how the text message is used to construct the class (here either *Task* or *Solution*).

The interface of the main class *Task* is composed of the following three functions: *Task()* - class constructor, *Solution createSolution()* - function generating an initial solution which can be either randomly drawn or empty, *TaskOntology ontology()* - function returning task ontology.

The interface of the main class *Solution* is composed of the following functions: *Solution()* - class constructor, *Solution(Task t)* - constructor producing solution to the given task, *SolutionOntology ontology()* - function returning solution ontology, *Object clone()* - function producing copy of the object, *boolean equals(Solution s)* - function returning the result of comparison between two solutions, *void evaluate()* - procedure evaluating the fitness of a solution (stored as a value of the variable *fitness*).

To obtain a solution of the particular problem instance the following actions should be carried out:

- Defining own classes *MTask* and *MSolution* inherited from *Task* and *Solution*, respectively. In these new classes constructors and other functions need to be over-riden to assure compatibility between actions and the problem instance requirements.
- Defining own ontologies *MTaskOntology* and *MSolutionOntology* inherited from *TaskOntology* and *SolutionOntology*, respectively. Both are responsible for translating classes *MTask* and *MSolution* into text messages. Messages are more complex than a single task or solution but to produce them the outcome of *MTaskOntology* and *MSolutionOntology* class functions are used.
- Defining auxiliary classes and functions as, for example, the *Compare* function, which could be used by the *SolutionManager* to compare and sort thus far obtained solutions.

JADE-Based A-Team environment includes a number of objects shown in Fig. 1. In this paper the focus is on optimization agents.

3 Optimization Agents

To better illustrate the role of the above described elements, this section focuses on an *OptiAgent* actions upon receiving the *OPTIMIZE* command with respect to a list of solutions (individuals) to a particular problem instance. An *OptiAgent* is brought to life to deal with a predefined type of task through applying certain improvement algorithm to obtain the improved solutions. Such an agent (for

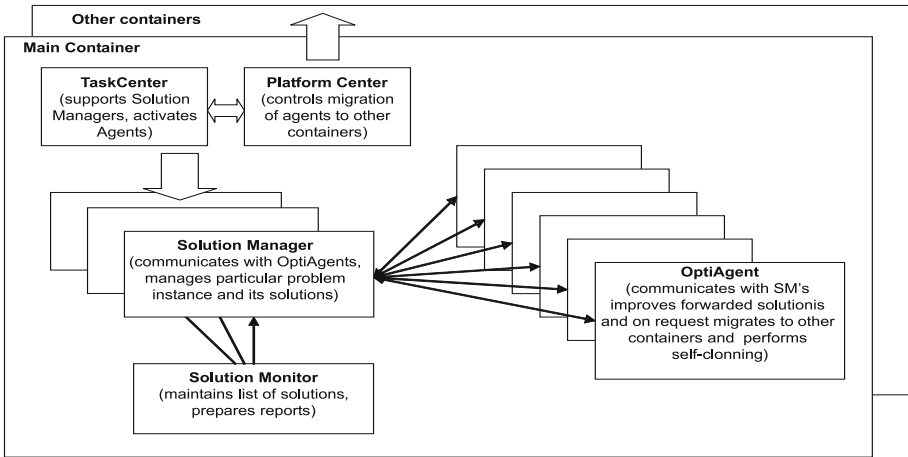


Fig. 1. JADE-Based A-Team environment

example the *TSPOptiAgent* dealing with traveling salesman problem instances) over-rides three important functions from the *OptiAgent* interface:

- *taskOntology()* - a function returning class, which defines ontology of the given task type.
- *solutionOntology()* - a function returning class which defines ontology of the solution for the task with the given *TaskOntology()*.
- *optimise()* which is a particular improvement algorithm for the given task type.

Responsibility for carrying out *OPTIMIZE* command stays with the *OptimizeSolution* class, which is a private class of each *OptiAgent*. The class is an extension of the cyclic action class available within JADE. In each cycle an *OptiAgent* reads the forwarded message (if any), processes it using the *optimize()* function and sends back the results, announcing also readiness for further action.

Example 1: Interface of the Data class

```

Task t;
ArrayList ss;
// constructors
public Data() {}
public Data( Task t, ArrayList ss) { }
// ontology functions
public Task getOpttask() { }
public void setOpttask( Task x) { }
public jade.util.leap.ArrayList getOptolutions() { }
public void setOptolutions(jade.util.leap.ArrayList l) { }
    
```

Receiving the forwarded message and resending the response requires two public classes - *Data* and *DataOntology*. *Data* is a public class. Its interface is

shown in Ex. 1. It consists of two fields representing a task and a list of solutions, constructors and several functions required to use JADE's ontology mechanism. These are *setX* and *getX*, where *X* is defined in the *DataOntology*.

DataOntology defines the structure of a message and binds its parts with relevant classes. A message is a command named *OPTIMIZE*, bounded to the *Data* class and divided into *OptTask* and *OptSolutions*. The *OptTask* part is bounded to the class representing a task and the *OptSolutions* part is a list of solutions. Each solution is bounded to the class representing a single solution. The code of the discussed fragment of the *DataOntology* is shown in Ex. 2.

Example 2: Fragment of the Data ontology

```
// vocabulary
public static final String OPTIMIZE = "Optimise";
public static final String OPTIMIZE_SOLUTIONS = "OptSolutions";
public static final String OPTIMIZE_TASK = "OptTask";
// schemes
// ontology sets values of the Data class fields
add( new AgentActionScheme(OPTIMIZE), Data.class);
// add schemes of the solution ontology s and the task ontology t
s.addSchemes( (Ontology) this);
t.addSchemes( (Ontology) this);
// solutions field of the Data class is a list of solutions
// with structure defined by s
as = (ConceptScheme)getScheme( OPTIMIZE);
as.add( OPTIMIZE_SOLUTIONS, (ConceptScheme)
    s.getScheme( SolutionOntology.SOLUTION),
    0, ObjectScheme.UNLIMITED);
// task field of the Data class has the structure defined by the t
as.add( OPTIMIZE_TASK, (ConceptScheme)
    t.getSchema( TaskOntology.TASK));
```

The above definition does not contain a detailed task and solution descriptions. These are taken from *t* and *s* ontology prepared for each instance of task and solution. It should be observed that when *DataOntology* is constructed both *t* and *s* are already known. Each *OptiAgent* has been brought to life with a view to optimize a particular task, which implies defining *t* and *s* which, in turn, can be accessed using functions *TaskOntology()* and *SolutionOntology()* from the interface of the *OptiAgent* class.

Finally, the code of the *OptimizeSolutions* class is shown in Ex. 3.

Example 3: Class OptimizeSolutions

```
private class OptimizeSolutions extends CyclicBehaviour {
    public void action() {
        ACLMessage msg = receive();
        if (msg != null) {
            try {ContentElement ce=getContentManager().extractContent(msg);
```

```

Concept cc = ((Action) ce).getAction();
if ( cc instanceof Data) {
    // the forwarded data are stored in the field o_data
    o_data =(Data)cc;
    optimize ();
    sendSolutions( msg.getSender()); }
} catch( CodecException ce) { ce.printStackTrace(); }
catch( OntologyException oe) { oe.printStackTrace(); }
};
ready(); //procedure of announcing agent readiness
block( 1000); // blocking agent for a period of time
}
}

```

4 Creating Agents

The proposed JADE-A-Team allows to create a variety of optimization agents searching, in parallel, for improved solutions to instances of one or more problem types and using various improvement algorithms. In Ex. 4 the operation of creating and activating an optimization agent and a solution manager is shown (in a working system there would be more agents of both types).

Example 4: Creating and activating an TSPOptiAgent and a SolutionManager

```

AgentController a=c.createNewAgent("AgentOpti1",
    "ASOP.TSP.TSPOptiAgent", null);
a.start();
Task z = new TaskTSP( "D:\\tasks\\TSP\\task1.txt");
// task will be forwarded to the SolutionManager as a parameter,
// preparation of the one-element list of parameters
Object[] _args=new Object[1];
_args[0]=(Object)z;
AgentController a=c.createNewAgent("SolutionManager",
    "ASOP.SolutionsManager", _args);
a.start();

```

The example *TSPOptiAgent* deals with improving solutions to the traveling salesman problem (*TSP*). The *TSPOptiAgent* class is defined in such a way that it uses a particular task type (here *TaskTSP* class), a particular task solution (*SolutionTSP*) and one particular improvement algorithm for optimizing.

The *SolutionManager* is created with a view to solving an instance of this particular *TSP* problem, which is send as a parameter. However, this class does not need to "know" that the task is an instance of the *TSP*. It will simply call relevant functions defined in main classes interfaces. For example, to create an initial solution manager can call the function *Solution createSolution()* from the interface of its task parameter.

The *TSPTask* instance parameters are read from the file *task1.txt*. *TSPTask* class has all the required objects defined including ontology, which can be used in the *DataOntology*.

In the present version all agents are created and activated to live a single and unique life by a special agent called *TaskManager*.

5 Managing Communication Between Agents

Solution manager, responding to announcements knows precisely to which *Opti-Agent* its message should be forwarded. Communication in the opposite direction is not that simple since optimization agents do not know which *SolutionManagers* operate on compatible tasks and solutions. To solve the problem a "yellow pages" service mechanism available in JADE has been used. It is provided by an agent called DF (*Directory Facilitator*) available in every FIPA compliant platform. Each *SolutionManager* provides the DF with its ID, the service name ("*solution management*") and the service type. In our case the service type is a label constructed from the names of ontologies of the task and solutions that the manager operates on. The respective part of the code is shown in Ex. 5.

Example 5: Registration of a manager in the yellow pages service

```
DFAgentDescription dfd = new DFAgentDescription();
dfd.setName( getAID());
ServiceDescription sd = new ServiceDescription();
sd.setType("solutions management");
// t is the task and s is a solution
sd.setName(t.ontologia().getName()+","+s.ontologia().getName());
dfd.addServices(sd);
try { DFService.register(this, dfd);
} catch (FIPAException fe) {fe.printStackTrace();}
```

Services of the *DF* are used by optimization agents to dynamically read the list of solution managers who could be a potential addressee of messages containing announcements of readiness to act. On the list, only solution managers offering "*solution management*" service for tasks and solutions with an ontology known to the optimization agent, are placed.

6 Conclusions

The proposed JADE-based A-Team environment is a "middleware plus" supporting development of A-Team systems. Its advantages have been inherited from JADE. The most important advantage which is preserved in the proposed JADE-A-Team is its ability to simplify the development of the distributed A-Teams composed of autonomous entities that need to communicate and collaborate in order to achieve the working of the entire system. A software framework that hides all complexity of the distributed architecture plus a set of predefined

objects are made available to users, who can focus just on the logic of the A-Team application and effectiveness of optimization algorithms rather than on middleware issues, such as discovering and contacting the entities of the system. It is expected that such an approach will result in achieving scalable, flexible, efficient, robust, adaptive and stable A-Team architectures.

During the test and verification stages JADE-A-Team has been used to implement several A-Team architectures dealing with well known combinatorial optimization problems. Functionality, ease of use and scalability of the approach have been confirmed. Further research will concentrate on providing a friendly human computer interface and developing a set of auxiliary agents that can be used to support the construction of dedicated A-Teams architectures.

Acknowledgement. The research was supported by the KBN, grant no. 3T11C05928

References

1. Aydin, M.E., T.C.Fogarty (2004) Teams of autonomous agents for job-shop scheduling problems: An Experimental Study, *Journal of Intelligent Manufacturing*, 15(4), p. 455-462
2. Bellifemine, F., G. Caire, A. Poggi, G. Rimassa (2003) JADE. A White Paper, *Exp*, 3(3), p. 6-20
3. Lerman, K. (2001) Design and Mathematical Analysis of Agent-Based Systems, J.L. Rash et al. (Eds.): FAABS 2000, Springer, LNAI 1871, p. 222-234
4. Marinescu, D.C., L. Boloni (2000) A component-based architecture for problem solving environments, *Mathematics and Computers in Simulation*, 54, p. 279-293
5. Parunak, H.V.D. (2000) Agents in Overalls: Experiences and Issues in the Development and Deployment of Industrial Agent-Based Systems. *International Journal of Cooperative Information Systems*, 9(3), p. 209-228
6. Rabak, C.S., J.S. Sichman (2003) Using A-Teams to optimize automatic insertion of electronic components, *Advanced Engineering Informatics* 17, p. 95-106
7. Rachlin, J., R. Goodwin, S. Murthy, R. Akkiraju, F. Wu, S. Kumaran, R. Das (1999) A-Teams: An Agent Architecture for Optimization and Decision-Support, J.P. Muller et al. (Eds.): ATAL'98, LNAI 1555, Springer, p. 261-276
8. Randall, M., A. Lewis (2002) A Parallel Implementation of Ant Colony Optimization, *Journal of Parallel and Distributed Computing* 62, p. 1421-1432
9. Talukdar, S., L. Baerentzen, A.Gove, P. de Souza (1996) Asynchronous Teams: Co-operation Schemes for Autonomous, Computer-Based Agents, Technical Report EDRC 18-59-96, Carnegie Mellon University, Pittsburgh

Agent Factory Micro Edition: A Framework for Ambient Applications

C. Muldoon¹, G.M.P. O'Hare², R. Collier¹, and M.J. O'Grady²

¹ Practice & Research in Intelligent Systems & Media (PRISM) Laboratory, School of Computer Science and Informatics, University College Dublin (UCD),
Belfield, Dublin 4, Ireland

{conor.muldoon, rem.collier}@ucd.ie

² Adaptive Information Cluster (AIC), School of Computer Science and Informatics,
University College Dublin (UCD), Belfield, Dublin 4, Ireland
{gregory.ohare, michael.j.ogrady}@ucd.ie

Abstract. Ambient Intelligence represents a vision of the future whereby the world will be saturated with embedded electronic devices that are sensitive and responsive to people. This technology will combine the concepts of intelligent systems with that of pervasive computing. Intelligent agents of varying capabilities will provide the foundations for many applications within this domain. As a means of achieving this objective a framework - Agent Factory Micro Edition (AFME) has been developed to enable the creation of agent-based applications on computationally constrained devices such as cellular digital mobile phones. It has been specifically designed to tackle the performance and memory footprint issues associated with executing intentional agents on mobile devices.

1 Introduction

Ambient Intelligence (AmI) represents the convergence of pervasive and intelligent systems, envisaging a world embedded with sensors and other electronic devices that communicate in a seamless and intuitive manner [1]. How to actually embed intelligence into computationally restricted artifacts and environments remains a key challenge. In an effort to address this issue a framework - Agent Factory Micro Edition (AFME), which supports the deployment of intelligent agents in AmI environments, has been developed. AFME is broadly based upon a preexisting framework that supports a structured approach to the development and deployment of agent-oriented applications, namely Agent Factory [2] [3].

Traditionally, applications incorporating Agent Factory have been deployed in workstation environments. The framework was implemented using Java 2 Standard Edition (J2SE). AFME differs from the original version in several ways. Many of these differences are because the system is based on the Constrained Limited Device Configuration (CLDC) Java platform augmented with the Mobile Information Device Profile (MIDP) rather than J2SE. CLDC and MIDP constitute a subset of the Java 2 Micro Edition (J2ME) Java specification.

A significant difference between (1) AFME and the original version of the framework, and (2) AFME and other embedded platforms, concerns the style in

which the object-oriented components have been written. Specifically, AFME has been developed without the use of accessor (get/set) methods. Accessor methods are really just an elaborate way of exposing an object's internal state and writing code in an alternative style considerably improves the maintainability of the software. It prevents alterations to the types of an object's internal attributes from propagating throughout the code. By minimizing code duplication the footprint of the system is reduced: a critical issue in embedded systems. In [4] the authors prove that any object-oriented system can be rewritten without the use of accessors¹ or exposing an object's state. Thus the developer is assured that there is always an alternative to the accessor approach.

AFME provides support for agnostic communication. The components of applications developed using the framework interact without directly referencing each other and are prevented from containing inter-component dependencies. This coerces developers into adopting a structure to their code that promotes reuse and modularity.

2 Related Research

Considerable research has been invested into the development of Multi-Agent Systems that operate on mobile devices. 3APL-M [5] is a platform that enables the fabrication of agents using the Artificial Autonomous Agents Programming Language (3APL) [6] for internal knowledge representation. Its binary version is distributed in J2ME and J2SE compilations. 3APL provides programming constructs for implementing agents' beliefs, goals, basic capabilities, and a set of practical reasoning rules.

The Foundation for Intelligent Physical Agents (FIPA) is an autonomous standards committee with the objective of facilitating interoperability among agent frameworks. It has ratified an international Agent Communication Language (ACL) to support inter-agent communication. MicroFIPA-OS is a minimised version of the FIPA-OS agent toolkit developed for mobile devices [7]. It provides support for the ACL standard along with yellow and white page services². The system can run in minimal mode whereby agents don't use task and conversation managers. The platform is entirely embedded; however it is recommended that only one agent should operate on low specification devices.

The Light Extensible Agent Platform (LEAP) [8] is a FIPA compliant agent platform capable of operating on both fixed and mobile devices with various operating systems. LEAP extends the Java Agent DEvelopment Framework (JADE) with a set of profiles that allow it to be configured for various Java Virtual Machines (JVMs). The platform is modular and contains components for managing the life cycle of the agents and controlling the motley of communication protocols. The platform is split into several agent containers - one for every device or workstation used.

Though sharing the same broad objectives of these projects, AFME differs in a number of ways. The system has been developed without the use of accessor methods

¹ The Law of Demeter is not limited to accessor methods. It is concerned with all methods that are not closely related to the object or class in question.

² Yellow pages enable agents to advertise the services they provide. White pages are used to find the address of an agent if the unique agent name is known.

or exposing an object's state. Writing code in an alternative style simplifies the message passing structure between objects, minimizes duplicated code, and in general reduces the footprint of the software. This is why AFME with a jar size of 86k is probably the smallest FIPA compliant deliberative agent platform in the world.

AFME differs from LEAP and MicroFIPA-OS in that AFME agents have the capability to reason about goals and intentions. Goals are essential for practical reasoning because they enable an agent to identify the purpose of a particular task. By abstracting this information we provide agents with a mechanism to recover from failures and to opportunistically take advantage of unexpected events or possibilities as they become available. Agents are resource bounded and will be unable to achieve all of their goals even if the goals are consistent. The subset of goals that an agent commits resources to achieving constitutes the agent's intentions. 3APL-M does provide rules for practical reasoning however AFME offers significant maintainability advantages over 3APL-M due to the object-oriented style in which it has been written and the support that it provides for agnostic communication (Section 5).

3 The Framework

AFME is based on Agent Factory, a preexisting FIPA compliant framework for the fabrication of a type of software agent that is: autonomous, situated, socially able, intentional, rational, and mobile [3]. It expresses an agent's internal state through the mentalistic notions of belief and commitment. Rules that define the conditions under which commitments are adopted are used to encode an agent's behaviour. This approach is consistent with the Belief-Desire-Intention (BDI) model of agency [10]. The framework is comprised of four-layers that deliver: a programming language, a run-time environment, an integrated development environment, and a development methodology. A detailed description of these components may be found in [3].

The differences between the specifications of the J2SE and J2ME have had a major impact on the design of the object-oriented components of AFME. A complete reengineering of the original system was necessitated, as it contained dependencies on APIs that do not exist within either CLDC or MIDP.

Although there are significant differences in the infrastructure used to build the platforms, AFME and the standard version of the system are consistent in terms of their support for executing agents written in the Agent Factory Agent Programming Language (AFAPL). Communication on both systems is FIPA compliant and thus interoperable. Agents on an AFME platform can migrate to a standard platform and vice versa. This consistency enables the developers of AFME applications to use the preexisting integrated development environment, methodology, and compiler for the creation of agent designs for constrained devices. These agent designs are used by the AFME compiler to generate the requisite MIDlet for the application.

4 Managing Complexity in AFME

Software systems have a greater degree of complexity for their size because if two parts of a program are the same they are placed within a subroutine [10]. Thus each

part of a computer program will be unique. In this respect, software differs profoundly from all other types of human construction in that it does not contain any repeated elements. Avoiding the use of accessor methods makes this ineluctable complexity of software easier to manage by reducing the entropy or restricting the movement of information within the system. It prevents the code dropping back to a semi-procedural system in which the developer has global access to information and an object's attributes move around a number of classes. Furthermore, maintainability is increased as an object's encapsulation is not violated, the types of its attributes are not exposed, and its internal structure remains hidden [11].

When writing code the programmer should view objects as a group of cooperating entities capable of performing tasks and passing messages to one another rather than as a data structure that contains functions or methods to alter its internal state. Objects should be conceptualized in terms of their capabilities and thus the focus of design should be on the messages that are passed between objects, not on how their internal structure has been implemented. The user of an object should never ask for data to perform some action; rather it should ask the object to do the work on their behalf [11]. The consequences of this are that messages flow within the system not data.

To see the negative impact that accessor methods have on an object-oriented system consider a class with a *getAttribute()* method that returns an *int*. Imagine this *getAttribute()* method was called 500 times by external users of the class. If at a later stage a seemingly trivial alteration were made such as changing the attribute's type from an *int* to a *long* a significant amount of code would have to be rewritten. This is because the developer would be coerced into making additional modifications at numerous locations where the method was invoked. Conversely, if the object had been designed to enable the external users to ask the object to perform the relevant work on the attribute on their behalf, this alteration and any related bugs would have been localized. The developer would only have to make a small number of modifications rather than hundreds of them.

Writing code in this manner has implications for how plug-ins may be developed. Classes that use accessor methods when extending the functionality of other classes are not really plugins in the true sense of the word. Consider a television whose functionality has been extended by a video recorder. If at a later stage the television's internal attributes are altered, for example a new flat screen television was bought, the video recorder does not have to be rebuilt. This is because it does not contain any dependencies on the internal components of the television. Now imagine that someone opened up the original television and used some of its parts in building a different device such as a microwave oven. When the television is replaced the microwave has to be rebuilt if it is to reflect the new changes of the television's internal attributes. What matters with plug-in technology is that the communication structures between the object and its extensions remain consistent. Any alteration to the internal attributes of the object or the extension is localized because these entities are only concerned about how they communicate with each other. When developing code that needs to be extensible, the manner in which it communicates with the plug-ins must be explicitly designed into the object. The video recorder could not have extended the television if the television were not specifically designed to have a socket for the video cable or the cables of other peripheral devices that communicate consistently. Otherwise the video recorder would have to access the internal components of the television.

When developing multi-agent systems for embedded devices we attempt to condense the same amount of functionality of the platforms developed for desktop machines into a much smaller space. This increases the complexity of the software for its size. Avoiding the use of accessors enables a developer to manage this increased complexity in that the code is structured in a manner that prevents maintainability problems. Accessor methods effectively make an object's internal attributes global therefore any alterations to the attributes will also be global. This causes precisely the type of maintainability problem that object-oriented programming is supposed to prevent. One of the tenets of good object-oriented development is that it is possible, by protecting an object's internal state, to radically alter the internal structure of a class without making changes to the users of the class. The developers of other embedded agent platforms that extensively use accessors have not adopted this concept. In contrast, AFME does not contain accessor methods³. It has been our experience that developing in alternative style usually improves performance because in most cases the number of method invocations within the system is reduced.

5 Supporting Agnostic Communication

AFME delivers support for the creation of BDI agents that follow a sense-deliberate-act cycle. To facilitate this process a number of system components have been created. Developers extend these components when building their applications. These components are perceptors, actuators, modules, and services. Perceptors and actuators enable agents to sense and to act upon their environment respectively. Modules represent a shared information space between actuators and perceptors. They are used when actuators and perceptors must communicate with an object instance internal to the agent, for example an actuator may pass a message to affect the state of an object instance and a perceptor may perceive that resultant effect. Services also represent a shared information space but between agents rather than actuators and perceptors. Services are used when information is passed between agents for example the local Message Transport Service.

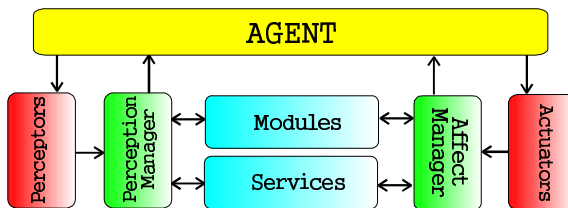


Fig. 1. Communication Structure of system components in AFME

To improve reuse and modularity within the system these entities are prevented from containing direct object references to each other and the agent class. Rather than passing messages directly they interact via perception and affect managers (Fig. 1).

³ No accessor methods have been written in the application code. Accessors are invoked on objects that form part of the core Java APIs where no alternatives are provided.

This ensures that communication between the components is agnostic. The messages that are passed between the components are in the form of first order structures. First order structures provide a symbolic representation of the information content and ensure that messages passed do not expose internal details of the message senders. Actuators and perceptors developed to interact with a service in one application can be used without making any coding alterations to interact with a module in a different application and vice versa. The implementation of modules or services can be completely altered without having to modify or recompile the actuators and perceptors. A completely different class could even be used to provide the functionality. Additionally, the same service or module may be used within two different applications to interact with a different set of actuators and perceptors.

The system components of AFME are interchangeable because they interact without directly referencing one another. They contain dependencies on the first order structure class and the affect and perception managers, which are generic components of the system. They do not contain dependencies on each other. When a module or service is created they are associated with a uniquely identifiable name. Actuators and perceptors use this name to indicate the target object for a particular message. They call the appropriate method on the affect and perception managers. The name is resolved to a module or service instance and the message is forwarded on appropriately.

6 Message Transport Service

The Message Transport Service of AFME (Fig. 2) had to be changed considerably from the original design. This was because our local GPRS and 3G service providers have a firewall operating thus preventing incoming socket connections and because MIDP and J2SE support different APIs for networking. Rather than having a server operating on the mobile device the message transport service periodically polls a mailbox server operating outside the firewall domain. Incoming messages are stored

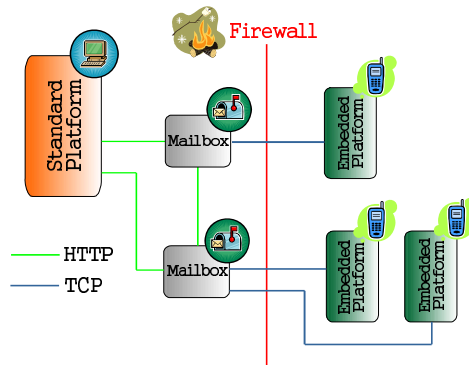


Fig. 2. AFME Message Transport Service

in the mailbox until a connection is made from the client devices, at which point all stored messages are transferred. This increases the latency of message passing but is necessary to pierce the firewall.

7 Migration

Within AFME support is provided for weak migration. Any classes required by the agent must already be present at the destination. This is because CLDC does not contain an API for introspection⁴ and is prevented from dynamically loading foreign objects. To facilitate the migration process a similar approach has been adopted to that of the Message Transport Service. Agents migrate to a migration server where they wait for a connection from their destination. When a connection is received they are transferred accordingly. When migrating back the agents also go through the migration server. As agents move to and from embedded devices the commitment rules that govern the agents' behaviour are altered to enable the agents to adapt to their environments. The agents' designs are decoupled into the core behaviours that operate on all platforms and platform specific behaviours. Agents maintain beliefs about where they can download the requisite commitment rules for a particular environment. When an agent migrates it sends this information to the migration server before it is transferred. When the destination platform connects to the migration server its type is specified. The migration server uses the destination type and the information sent by the agent to obtain the requisite commitment rules. The rules are then added to the agent design. This enables a transparent migration process. The agent need not be aware of the type of environment it is migrating to before it migrates.

8 Conclusion

Intelligent agents encapsulate certain characteristics that make them suitable for creating ambient applications. Their autonomous nature, ability to react to external events as well their capability to be proactive in fulfilling their objectives make them apposite for operating in complex and dynamic environments. Embedded devices are, almost by definition, computationally constrained. Thus the goal of delivering intelligent agents on such devices is one fraught with difficulty, since agent platforms often have a large footprint. AFME is a framework that been developed to address this issue. The design decisions taken for the construction of the platform have been described in this paper. The approach taken significantly improves the maintainability of the software when compared to other embedded agent platforms currently available.

Complex software solutions will be required if the vision of Aml is to be fulfilled. Embedded intelligent agents offer one promising approach to realising the *intelligence* essential to Aml. Through the development of AFME we have demonstrated that such

⁴ Local classes packaged within the application jar file can be dynamically loaded.

an approach is feasible. AFME is an open source project and is freely available for download from the Agent Factory SourceForge web site under the terms of the GNU Lesser General Public License [12].

References

1. Aarts, E, Marzano, S. (editors), *The New Everyday: Views on Ambient Intelligence*, 010 Publishers, Rotterdam, The Netherlands, 2003.
2. O'Hare G.M.P., *Agent Factory: An Environment for the Fabrication of Multi-Agent Systems*, in *Foundations of Distributed Artificial Intelligence* (G.M.P. O'Hare and N. Jennings eds) pp. 449-484, John Wiley and Sons, Inc., 1996.
3. Collier, R.W., O'Hare G.M.P., Lowen, T., Rooney, C.F.B., (2003), *Beyond Prototyping in the Factory of Agents*, in *Multi-Agent Systems and Applications III: Proceedings of the 3rd Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'03)*, Prague, Czech Republic, Lecture Notes in Computer Science (LNCS 2691), Springer-Verlag.
4. Lieberherr, K.J., Holland, I, Riel, A.J., *Object-oriented programming: An objective sense of style*, in *Object Oriented Programming Systems, Languages and Applications Conference*, in special issue of SIGPLAN notices, number 11, pages 323-334, San Diego, CA, 1988.
5. 3APL-M: Platform for Lightweight Deliberative Agents: <http://www.cs.uu.nl/3apl-m/>
6. Birna van Riemsdijk, M.D., Dignum, F., Meyer, J.J., *A Programming Language for Cognitive Agents: Goal Directed 3APL*. *Proceedings of the First Workshop on Programming Multiagent Systems: Languages, frameworks, techniques, and tools (ProMAS)*, Melbourne, 2003.
7. Tarkoma, S., Laukkanen, M., *Supporting Software Agents on Small Devices: Proceedings of the first international joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Bologna, 2002.
8. Berger, M., Rusitschka, S., Toropov, D., Watzke, M., Schichte, M., *Porting Distributed Agent-Middleware to Small Mobile Devices: Proceedings of the Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices held in conjunction with the joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Bologna, 2002.
9. Rao, A.S., Georgeff, M.P.: *Modelling Rational Agents within a BDI Architecture*. In: *Principles of Knowledge Representation. & Reasoning*, San Mateo, CA. 1991.
10. Brooks, F.P., *No Silver bullet: Essence and Accidents of Software Engineering*, in *IEEE Computer*, Vol. 2 No. 4, April 1987, pp. 10-19.
11. Holub, A., *Building user interfaces for object-oriented systems*, Part 1.
12. http://www.javaworld.com/javaworld/jw-07-1999/jw-07-toolbox_p.html
13. *Agent Factory SourceForge repository*. <http://agentfactory.sourceforge.net/>

Crises Management in Multiagent Workflow Systems

Małgorzata Żabińska

Department of Computer Science, AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Kraków, Poland
zabinska@agh.edu.pl

Abstract. High degree of complexity of processes connected with compound chains of operations requires search for new models and methods. Moreover, one should take into account possibilities of presence of undesired situations, called crises, when system functions in real conditions. The aim of the work is to create a model of a system based on application of agents, which manages a process of realization of chains of operations (either a "matter" to be processed or production process) and its structure in such a way that it is possible to synchronise the model by real process. In the paper, a concept of multiagent system solving this problem, concerning potential crisis situations, as well as prototype realisation for a sample application have been shown.

1 Introduction

Methods of supporting processes of automatic management of chains of operations: either production (assembly line) or related to flow and processing of documents (document workflow), have been worked out for many years. Different practical solutions on the basis of workflow systems [5] have been proposed, but none of them solves all related problems which may appear in process of business or production enterprise management. Moreover, existing workflow solutions have disadvantages. The first of them is that they are static: workflow systems use some fixed model of a system, resulting from future system work conditions. Potential change of conditions causes limitation of usefulness of a model, and thus – a system, as well. Therefore it requires repetition of analysis and change of a model, which implies new outlays. Especially essential and difficult problem is potential appearance of undesirable situations, the so-called "crisis situations", during realisation of operations chains. There exist a group of crises, which can be considered at design of the system, especially when applying modern concepts, e.g. agents' approach, i.e. creating a system of management on the basis of multiagent system. It seems that just a concept to merge agent systems with workflow systems can enable removal of at least some of the above-mentioned disadvantages, by assurance required system dynamics and flexibility connected with real conditions of management process of operation realization. The aim of the work is to create a model of the system based on application of agents. It should manage both: a process of realization of chain of operations ("matters" to be processed related to flow and processing of documents or similarly – production process) as well as structure of the process, in such a way, that synchronization of the model by a real process is possible.

2 A Concept of Multiagent System for Management of Chain of Operations by Example of Flow of Documents

The basic idea of workflow system is functioning as an assembly line with multiple posts which act as narrowly specialised service points. These posts may supply the same, similar or completely different services. In case of a system for management of chains of operations on documents, input documents contain, among others, information on a procedure which they are to be subjected, i.e. a list of types of posts, which they should visit for processing (performing appropriate operations on them). On the basis of this information, documents are directed to proper service points.

In workflow systems based on application of agents, these agents support and coordinate processes making a workflow. They move within a system according to defined principles. During performing operations, they may carry resources of the system with themselves (e.g. data: documents or descriptions of procedures), reducing at the same time the need to access common resources. Various types of agents related to different types of tasks in the system, are applied. Usually a single workflow agent is responsible for control of a single process. Mobile Tasks Agents, which migrate to places where these tasks are performed, are singled out, as well as Personal Agents, which make interface between agents and the part of the system for task realization. There are also such approaches, where more types of agents are present, that are connected with their roles in the system as more detailed and individual assignment of tasks ([4]). Nevertheless in the existing approaches, potential crisis situations related to real conditions of functioning of a system serving process of chains of operations, have not been considered.

The presented concept of multiagent system for management of process being a chain of some operations (on documents or technological ones), takes into consideration crises and comprises essential details presented below. The main element of the concept consists in representation of a structure of an enterprise as a graph [6]. Fundamental elements of the system are objects, which are subjected to processing in production process or in process of document processing during performing some "matter" – related to realization of some operations on them. These elements are represented by agents choosing optimal path in a graph according to a given criteria (i.e. the best one in the defined sense: the cheapest, the most expensive, the quickest, etc.). Nodes of the graph representing operations performed on these elements (agents) are also managed by a system, which consists of agents of the appropriate type (performing these operations or collecting information and proposing changes in the graph structure).

Agent of the first type (Task Agent or *Document Agent*) is a basic element of the system and possesses its own description of operations (e.g. in the form of own graph), which have to be performed on it. It has also a given strategy of choice (e.g. the quickest, the cheapest, by the given date). Agent-element of the system pays for each operation (by means of real transfer of currency, with counters or otherwise). It makes the fundament for market decisions on further route. In case of a system for service of documents, such an agent is created at the moment of introduction of a document into the system and it migrates with it through appropriate posts in the system. The agent's task is to choose a route of a document through these posts, and it

is connected with the certain document until the end of its processing, i.e. until removal from the system.

Nodes of the graph, storing transferred means, make up market decisions concerning development (here also: creation) or reduction (also: liquidation) of connections. Agents of the second type (*Post Agent*) connected with graph nodes are used for this purpose and they negotiate as well, cooperation concerning realization of changes in the graph. For a system of service of documents, the agent assigned to a graph node (post of service) should realize the service in the node and make information about the node available (service characteristics). The latter one makes a basis for choice of a route (a node i.e. a post) by an agent of the first type (agent-element of the system).

Additionally, the other agent – of the third type (*Information Agent*) should collect information on system functioning and propose changes in the structure of the system. We can also anticipate an agent of the fourth type (Monitoring or *Supervisor Agent*), which controls the work of the whole system and on the basis of information collected from subordinate agents, decides on potential changes in the structure of the system (maybe necessary because of presence of some crisis situation). Functions of agents of the third and the fourth type can be merged in one agent or split between two of them. In the concept of the system for service of operations chains (on the example of processing of documents) taking into account potential crises described in the next chapter, main types of above-mentioned agents have been considered, as well as existence of additional specialized agents. As for latter ones, there may be e.g. three types of agents, related to division of their auxiliary roles in the system: *New-Tasks Agent*, *Revising-Tasks Agent*, *Waiting-Tasks Agent*. Two former ones perform roles of *Personal Agent*, having possibility of communication with the user of the system. A model of communication in such a system, i.e. between the user and agents, as well as agents themselves, is shown in the diagram (Fig. 1.).

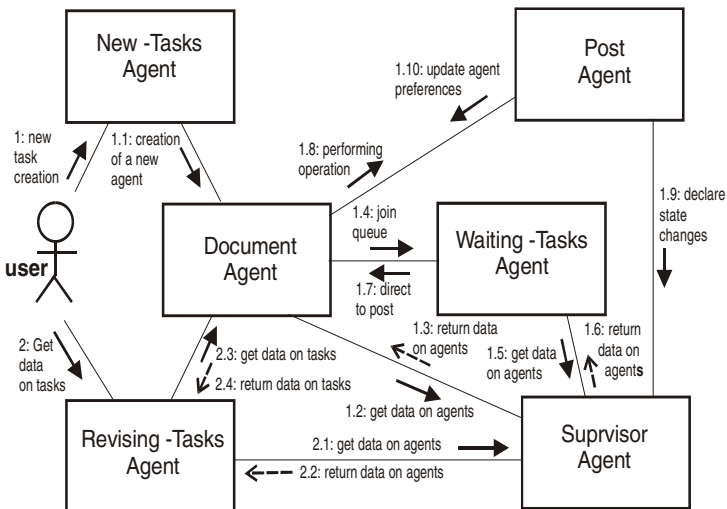


Fig. 1. A concept of communication in the proposed multiagent system

3 Crisis Situations and Their Service in the Proposed Multiagent System

During real work of a system for management of process of operations chains realization, a number of undesirable situations may occur. Most of them can be considered during creation of a model of a system, especially when designing a solution as a multiagent system. A short description of the most typical crisis situations, with proposals of solutions in the system based on agent approach is presented below. Selected UML sequence diagrams show activities to be performed by the system to solve chosen crisis situations. Unfortunately, some crises require intervention of a human-being (e.g. system administrator), but they are fortunately less probable.

A post (Post Agent) is overloaded – a queue is too long (1). Such a situation may often occur. The simplest solution is to assign the other agent, which either takes over all tasks (in case of total inoperability of a post), or only some part of them, such as to unload a congestion (to shorten a queue). Supervisor Agent is responsible for assignment of a new Post Agent to support the post. However the former one should be informed about the situation by Information Agent. A model of solution of crisis situation (1): overloaded post (a queue is too long) is shown in Fig. 2.

A post (Post Agent) idle – null queue of tasks (2). This crisis situation is recognised on getting data about agents (Post Agents) by Supervisor Agent. Thus each time when some agent (Document Agent) wants to know something about certain post or other agents, Supervisor Agent finds out idle state of a post and may remove it from the structure. In case of Post Agent which has been idle for a long time, it is assigned by Supervisor Agent as the first one to be used for service of crisis situation (1): too long queue (overloaded post). When a great number of unused Post Agents appears, it is possible to remove some of them and create one agent responsible for all tasks, which have belonged to liquidated agents – predecessors.

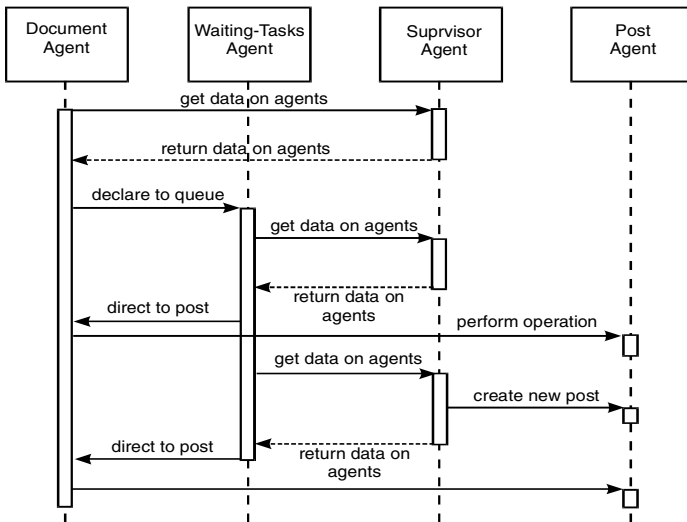


Fig. 2. Sequence diagram for service of crisis situation (1): overloaded post

Document (Document Agent) is lost (3). This case is difficult to be modelled, because the moment of the loss (or its recognition) is usually not known. Supervisor Agent should check each given interval of time, whether state of agents is appropriate and thus it may recognise the loss of a document. Each document is accompanied by Document Agent, which sends message to Information Agent when going through a post. When Document Agent is lost, Information Agent should have possibility of reconstruction of a document route and locating the point of loss. It should also discover whether a document (Document Agent) has been destroyed (killed), or whether trial to take over the data carried by the latter one has occurred (and inform Supervisor Agent about it). In the worst case, i.e. when reconstruction of the already processed data is impossible, such a loss implies direction of a document to repetition of processing. To serve such a crisis, one can plan auxiliary agents to convoy Document Agent (especially in environments of high level of threat). They take over the data at threat, or when it is not possible, they direct a task (document) to process it again (with Supervisor Agent, making the decision, as an intermediary).

Loss of Information Agent (4). Such a crisis situation has to be discovered by Supervisor Agent, which is responsible for creation of a new Information Agent. The latter one takes over all tasks of its predecessor. Sequence diagram modelling this crisis situation is shown in Fig. 3.

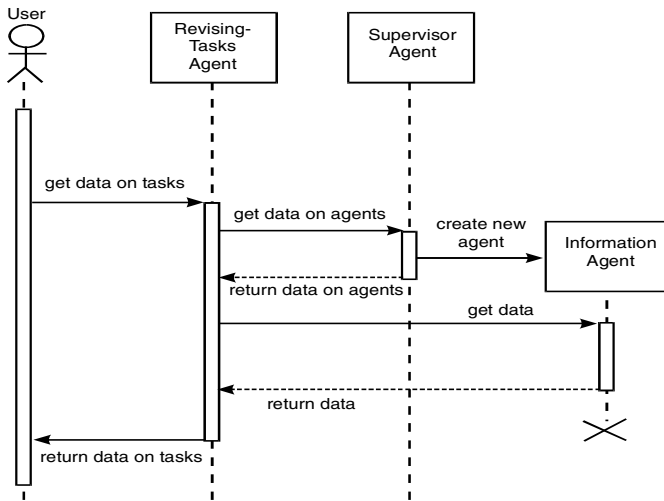


Fig. 3. Service of crisis situation (4): loss of Information Agent

Loss of Supervisor Agent (5). Such a type of a crisis makes continuation of system work impossible, whereas Supervisor Agent does not exist any more. Thus there is no agent monitoring the system as a whole, i.e. flow of information from Information Agent, collecting data from Posts Agents about posts' load, as well as data flow between Posts Agents. The solution is to create a new Supervisor Agent. It requires transferring a message (maybe coming from Information Agent) about crisis situation

and intervention of a human-being (e.g. System Administrator) to cause creation of a new Supervisor Agent (what may require restart of the system or at least its part).

4 Realization of Multiagent System for Management of Operations Chain by Example of Flow of Documents

Let us consider operations chain, e.g. some matter to be settled that is connected with flow of documents between posts, where documents are being processed. A proposal of realization (prototype version) of such a system with the use of agents [3] will be presented below with assumptions concerning a concept of multiagent system application: documents for processing are generated by the system; user gives a number of documents to be generated and intervals of time; user defines an arrangement of posts (a graph); posts may be of different type and have different characteristics within the type; there are characteristics of the post, viz. type, location, cost of the way to other posts, price, time and quality of service; the final (proposed) structure of the system (a graph) should be stored; possibility of crises shown in the previous chapter is considered.

The system comprises the following basic types of agents, shown in brief in a description of a concept: *Document Agent* – mobile agent, which accompanies a document during its "life" in the system; it collects information about available posts (which services they have, of what quality, for what price, what the waiting time for realisation of service is, price of the way between a document and the post), and it chooses posts for document service, as well; each document has its own agent. *Post Agent* – agent connected with a graph node; it realizes its service and makes information on its post available – what it does, with what quality, for how much and how long, what estimated waiting time for service is (the queue of documents for the post); each post has its own agent. *Information Agent* – the global agent of the system, which collects information from Posts Agents, generates reports on load of single posts, and proposes potential changes of the structure (i.e. the number of posts and their arrangement). *Supervisor Agent* – supervises the whole work of the system, communicates with Information Agent, and undertakes decisions on introduction of proposed changes of system structure (based on data delivered by the latter one).

The main task of Document Agent is to choose a sequence of posts to process the document, which the agent is connected with. Documents are introduced into the system and then they flow between posts realizing subsequent tasks on its way. Each document has its individual route, i.e. ordered list of nodes (appropriate processing posts), through which it should go and a priority which indicates how quickly it should finish its way through all posts (the higher priority, the more important the document is). Document Agent chooses the subsequent post (node of the graph) among posts performing certain service, which is planned for a document in the next step of processing procedure of the document.

For this purpose each post i.e. the node of the graph (or Post Agent) makes the following information available: type – a sort of service (integer number from 0 to 100); price – price of service (integer number from 0 to 100); time of service – measured in simulation steps (or virtual units of simulation time, up to 100); quality – service quality (integer number from 0 to 100); length of the current queue – number of tasks waiting for service (here: documents).

The next post is chosen by comparison of characteristics of considered posts. The following criteria are used: price, time length of a queue, cost of the route and quality – taking into account all posts of a given type. The value "result" is calculated for each post, which is considered:

$$\text{result} = \text{quality} + \text{price} + \text{distance} + (\text{priority} * (\text{time} * (\text{queue} + 1)))$$

If two or more posts have the same value, then only the period of time to the end of service of document is taken into account: $\text{time} * (\text{queue} + 1)$. If it is not enough, the following values: distance (route or cost, quality and price) are compared. Finally if posts are identical (as a result of comparison), then choice of the post is performed by drawing a post from the poll of posts left after the previous step. Documents leave the system after their service is finished at the last post from the list.

The main task of Information Agent is observation of system work, evaluation of structure of processing system and proposing changes of structure of workflow system. All the time during work of the system (or simulation), Information Agent traces realization of operations on documents – collects information from Post Agents and generates reports. They contain data on the current state of posts: id of agent, type, price, quality, length of queue, number of documents already having been served (from the beginning of simulation). On the basis of this information, changes of the structure i.e. removal or addition of posts, are made by Supervisor Agent.

For this purpose the following calculations are performed: real time of work is calculated for each Post Agent, as product of service time of the document and the number of served documents, divided by the total time of simulation (program run):

$$\text{work time} = (\text{service time} * \text{number of served}) / \text{simulation time}$$

The result is the value from range [0, 1]. On this basis, the number of posts of a given type, which should compose the system, is calculated as it follows:

$$\text{number of posts} = \text{sup}(\text{worktime}_1 + \text{worktime}_2 + \dots + \text{worktime}_N)$$

where N : number of posts of a given type (given group).

Procedure of change of the number of posts is the following: if the calculated number of posts is less than the current number of posts of a given type, posts are removed in order opposite to calculated value of time of post work, until the number of posts of this type (a new value) is reached. Next, groups of posts which have not decreased their cardinality and mean lengths of their queues were greater than zero, are checked. Then it is necessary to add a post or posts to a group. To do this, data concerning queues to posts, collected by Information Agent during work of the system, is used.

Behaviour of a queue is described by some value – a queue indicator, calculated as quotient of number of documents in the queue (a current length of the queue) and theoretical number of so-far served documents (i.e. number of documents served from the beginning of the run, on the assumption that the post has been working all the time). The theoretical number of served documents is calculated as quotient of the time from the beginning of simulation and the time of document service by this post. Therefore queue indicator is defined as it follows:

$$\text{queue indicator} = \text{length of the queue} * \text{service time} / \text{simulation time}$$

These queues' indicators are collected until the last document is introduced into the system. Next, their average value is calculated, summed up within the group and rounded to the integer value. It makes the number of posts to add to a group of posts of a given type. It is assumed that parameters of added posts are as parameters of the post, the most loaded in the group. The time of arrival of the last document to a group is the time of finishing simulation when there are no queues; thus when a satisfactory number of posts to serve all documents exist.

5 Conclusion

A concept of multiagent system for management of chains of operations (technological ones or performed on documents), which takes into account undesirable situations (called crises) has been presented in the paper.

Roles of agents connected with assignment of tasks in multiagent workflow system have been described. An outline of realization of a prototype multiagent system for management of documents flow and crisis situations, which may appear in real process composed of chain of documents processing operations has been shown.

The prototype system has been realized according to incremental model of software development. The current – first increment is related to creation of basic functions, comprising main agents and principal parts of the system. It should be enriched with the following elements: additional functions connected with auxiliary agents, service of rare crises, elements of optimization, and elements related to system security.

It seems that multiagent system of the proposed type can make processes of analysis and evaluation of existing workflow systems – effective, as well as the process of designing such systems – more efficient, by considering real working conditions, i.e. threat of potential crises.

References

1. Ambroszkiewicz, S., Cetnarowicz, K., Kozlak, J., Nowak, T., Penczek, W.: Modelling Agent Organizations. In: Klopotek, M., Michalewicz, M., Wierzchon, S. T. (eds.): Proceedings of the Conf. Intelligent Information System. Advances in Soft Computing, Springer Verlag Berlin (2000) 135-144
2. Cetnarowicz, K., Kozlak, J.: Multi-Agent System for Decentralized Computer Network Management. Management and Control of Production and Logistics, IFIP, IFAC, IEEE Conference, ENSIEG, LAG Grenoble, France (2000)
3. Cetnarowicz, K., Żabińska M.: Management of operations realization process based on multi-agent system, (in Polish), In: Proc. of the VIII Int. Conf. Management of Business Enterprise. Theory and Practice, AGH, Kraków (2005)
4. Stormer, H. and Knorr, K.: AWA – eine Architektur eines agentbasierten Workflow-Systems. In: Tagungsband 5. Internationale Tagung Wirtschaftsinformatik (WI 2001), Augsburg, Germany, (2001) 147–160
5. Workflow Management Coalition. Terminology – Glossary. Document Number WFMC-TC-1011, www.wfmc.org
6. Żabińska M., Cetnarowicz, K.: Multi-Agent Simulation Model for Control of Transport System, In: Štefan, J., (ed): Proceedings of 37 Int. Conf. MOSIS'03, ISBN 80-85988-86-0, MARQ, Ostrava, Czech Republic, (2003) 327-334

Agent Architecture for Mesh Based Simulation Systems

K. Banaś

Cracow University of Technology
Warszawska 24, 31-155 Kraków
kbanas@pk.edu.pl

Abstract. The paper presents an analysis of requirements for building simulation systems with tightly coupled components, such as typical mesh based PDE approximation software. The considered systems are characterized as having high communication to computation ratio. When designing architectures for such systems the hardware and middleware capabilities for providing communication links between processes have to be investigated and fully exploited. This is the place where agent technology perfectly fits the requirements. In the whole system, the capabilities of agents should be complemented with less flexible but more efficient software organization.

As an example a framework for finite element simulations, employing a modular architecture (described in [1]), is considered. Communication requirements for typical computations are estimated and evaluated in view of possible inter-process communication. The role of agents in setting up the execution structure of simulations is described.

1 Agent Based Simulation Systems

Agent technology is often used to add flexibility to classical computational systems that are executed on distributed hardware resources [2, 3]. Agents can be used to query static information on hardware resources, as well as to monitor their workload. Based on the data the optimal mapping of computations on processing elements together with proper communication patterns can be selected [4]. Autonomous agents can additionally gather necessary information and take decisions: concerning e.g. splitting computations, migrating or stopping. The current paper describes a setting where the above mentioned possibilities of agents are crucial for achieving the assumed goal. The setting consist of a classical computational science system, a mesh based PDE solver, for which a new flexible architecture, that can take advantage of modern hardware resources is sought.

2 Mesh Based Computations with Tightly Coupled Components

Mesh based computations, using the finite difference, finite element and finite volume methods, are common in scientific computing. Such computations contain, as an important ingredient, matrix operations, and therefore the focus of

research on such systems, in view of achieving high performance, is often put on developing linear algebra packages. However, the overall efficiency of such codes may depend on the architecture of the whole system, including the interaction of algorithms and data structures that deal with mathematical entities appearing in the problem domain: discretized computational domains, discrete fields and systems of algebraic linear equations. It is possible to develop software architectures that use different modules to deal with the above mentioned entities and to combine the resulting flexibility in setting up the computational system with high performance of execution. Such architectures can be useful when complex data structures and algorithms are used for e.g. adaptive mesh refinement, projections of complex discrete fields or multigrid solvers and preconditioners.

The agent technology for modular mesh based solvers can be employed through linking each module (component) with an agent that serves as a control and steering unit for the module. Such an agent has to know the specifications of hardware resources as well as the detailed characteristics of the modules it controls, including the required communication with other modules. This knowledge is used to properly configure a computational system, exploiting the capabilities of hardware and software components.

Next sections describe an example architecture for a mesh based solver that uses the adaptive finite element method. From the three, mentioned previously, numerical methods used in mesh based simulations, the finite element method is theoretically the most complex: it can use unstructured adaptive meshes and higher order approximation. It is also the one, that usually suffers from the most severe performance penalty due to the need to deal with the above mentioned complexities. Therefore it is important to design architectures for FEM systems that can take advantage of different, flexible hardware environments.

3 Modular Architecture for Sequential Finite Element Systems

Classical architectures for finite element codes usually distinguish between the two levels of coupling among their building components. At the loosely coupled level there are modules for three phases of simulations: pre-processing (creating the description of a computational domain, generating a mesh for the domain), processing (actual calculations) and post-processing (calculation of derived quantities, visualization). At that level the requirements for communication between modules are low and the modules are often organized into different programs exchanging data by means of files that are produced by one module and then subsequently read by another.

At the tightly coupled level there are submodules that handle specific data structures and computations within the simulation phases. In the sequel, only the phase of actual calculations will be considered in detail, as usually the most time consuming. For the component that realizes this phase, the finite element computational kernel, a modular architecture has been proposed [5], that aims at creating software more flexible and easier to modify and extend. This architecture,

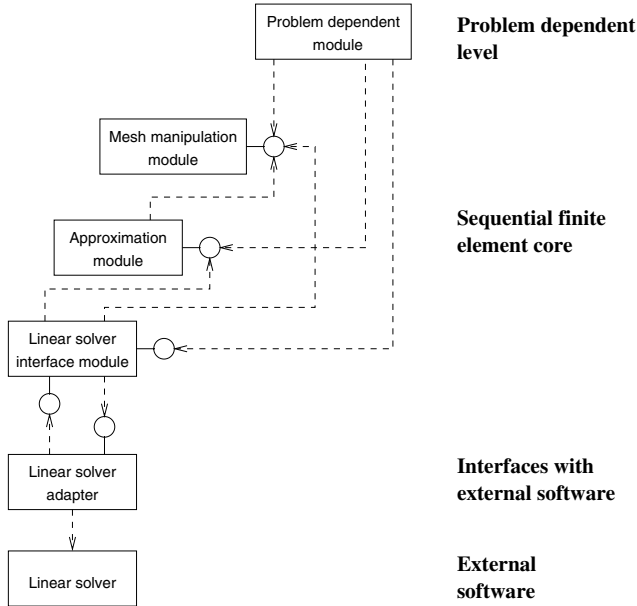


Fig. 1. The proposed architecture for sequential finite element codes

illustrated by a UML diagram in Fig. 1, consist of several modules, that are distinguished based on the data structures they operate on. The reason for this is that classical structural analysis of FEM computations, that finds modules by looking at the flow of computations, is no longer the only possible and even not the most appropriate, as the computations become more and more interactive. Hence there are the following modules and data structures in the proposed architecture:

- problem dependent module with algorithms and data structures particular to a problem solved
- mesh manipulation module with a data structure that holds all topological and geometric data on a mesh; algorithms in the module perform e.g. mesh modifications
- approximation module; the module responsible for storing discrete data on physical fields and for realizing the necessary operations on the fields (like e.g. integration, differentiation, projection)
- interface with linear solvers module that translates information expressed in terms particular to the PDE approximation methods into information expressed in terms of numerical linear algebra entities like vectors and matrices; this module can be equipped with additional data structures, e.g. for reordering linear equations
- linear solver which is considered an external software in the proposed architecture

4 Domain Decomposition for FEM Core

For parallel execution the extension shown in Fig. 2 has been proposed [1], that uses a centralized management based on domain decomposition and message passing. This architecture enables achieving high performance of execution for static parallel hardware configurations. However, from the software engineering point view it goes against the design goals of the sequential version. The idea to create independent components for dealing with fundamental data structures of the code does no longer holds.

Therefore in the current paper a natural next step in development of modular design for computational kernels is presented. This next step consist in equipping each sequential module of the code with its own domain decomposition manager. The architecture is shown in Fig. 3. The parallel modules, consisting of pairs sequential module–domain decomposition manager, can now be developed and tested separately. Additional flexibility is brought by the possibility of using the same domain decomposition managers for different sequential modules of the same kind and exchanging in a particular code only one parallel module at a time.

The standard run-time structure of a parallel code built according to the architecture consist of a set of processes executed on different processing nodes of a computational system. The processes correspond to compiled FEM programs, each of which contains all modules from Fig. 3.

It is known that with the increasing number of subdomains (processes) the parallel efficiency of fundamental algorithms, such as computing vector norms

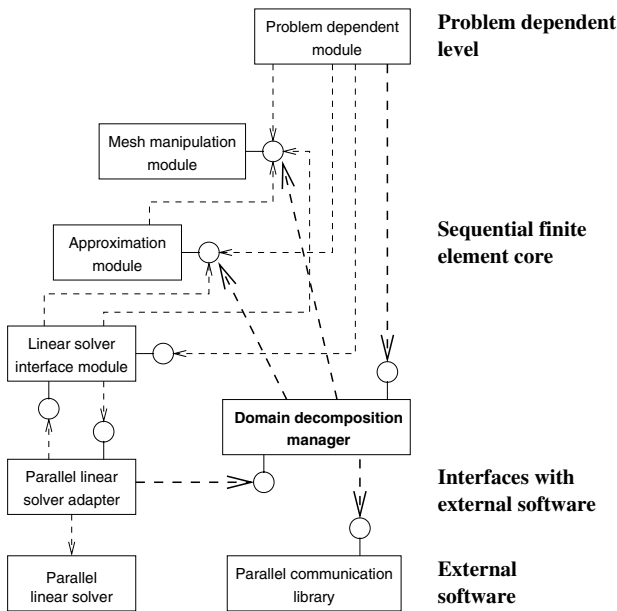


Fig. 2. The centralized architecture for parallel finite element codes

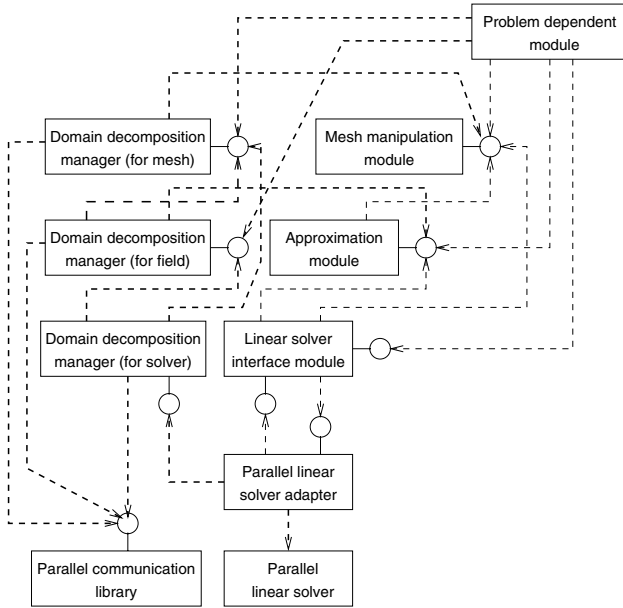


Fig. 3. The proposed modular architecture for parallel finite element codes

and scalar products or matrix-vector products, that form building blocks of mesh based simulation systems, diminishes. Moreover, the convergence of iterative solvers usually slows down. In order to maintain algorithmic complexity of sequential computations and to maximize utilization of computing systems comprising of more and more processors, different parallelization strategies, beyond simple domain decomposition and message passing, have to be employed. One of such strategies is to combine message passing with multi-threading [6]. Still the processes correspond to programs consisting of all modules, the difference lying in the multi-threaded mode of execution, using compiler directives or OpenMP.

5 Functional Decomposition for FEM Core

Further increase in flexibility of setting up a run-time structure of codes can be obtained by investigating the possibility of using separate processes for different parallel modules of the code. In such a case each dependence between modules should be now understood as requiring communication: each inter-module function call involves sending a message. This creates two kinds of communication links for each module: horizontal – among modules of the same kind managing a distributed data structure and vertical – between different modules.

It is interesting to estimate what are the communication requirements for different modules in the above mentioned situation. The analysis below uses simple and typical settings, more detailed investigations can be found in [7].

The main steps of the typical FEM solution process comprise numerical integration of terms from the finite element formulation of the problem, assembling of the linear system matrix and solving the system of linear equations. In the parallel execution model with domain decomposition and message passing the first two phases can be performed perfectly or almost perfectly in parallel. When typical iterative solvers are used for linear system solution phase, communication requirements are mainly related to matrix-vector products.

The communication complexity of performing matrix-vector products for FEM matrices depends on their non-zero structure. This structure reflects the whole finite element setting: the type of problem solved, the dimension of the physical space, the mesh employed, the kind of approximation. The most important ingredient (the other change the communication cost by constant factors) is domain decomposition that determines the ratio of the number of unknowns (degrees of freedom) inside domain to the number of unknowns on the boundary of domains. When performing matrix-vector products data corresponding to intersubdomain boundary has to be exchanged among processes. In the simple case of very regular 3D domains with perfect domain decomposition the number of exchanged unknowns per subdomain can be estimated as being in order of $(N/N_S)^{2/3}$, where N is the total number of unknowns and N_S is the number of subdomains. During the whole solution process the number of data exchanges is usually equal to the number of iterations, N_{it} , which practically is of order 10–1000.

Among the parallel modules exchanging data vertically the pair approximation module–linear solver interface module requires the least communication. The minimal estimate for this communication comprises sending all entries of the system matrix (this neglects, among others, sending an initial guess vector, a right hand side vector and the information on the structure of the matrix). The number of entries per subdomain, in the same case as considered for the communication analysis, is in the order of $(N/N_S)N_{nz}$ where N_{nz} is the average number of non-zero entries in a single row of the system matrix. The last number for typical problems, meshes and approximations is of order 10–1000.

Hence the amount of vertical communication is, in typical configurations, at least one order of magnitude greater than the amount of horizontal communication (the ratio of vertical communication to horizontal communication in the considered particular case is equal at least to $(N/N_S)^{1/3}N_{nz}/N_{it}$). Taking this into account the question arises, whether the vertical splitting of modules can bring performance advantages? This may take place for hardware systems consisting of several SMP nodes or, the solution that will become more and more popular in the future, multi-core processors. Fast communication using shared memory can be used to execute modules in parallel, keeping hardware resources busy, and not increasing the number of subdomains.

6 Agent Based Architecture

For the described architecture with horizontal and vertical splitting of execution components the proper mapping of computations on the hardware become a

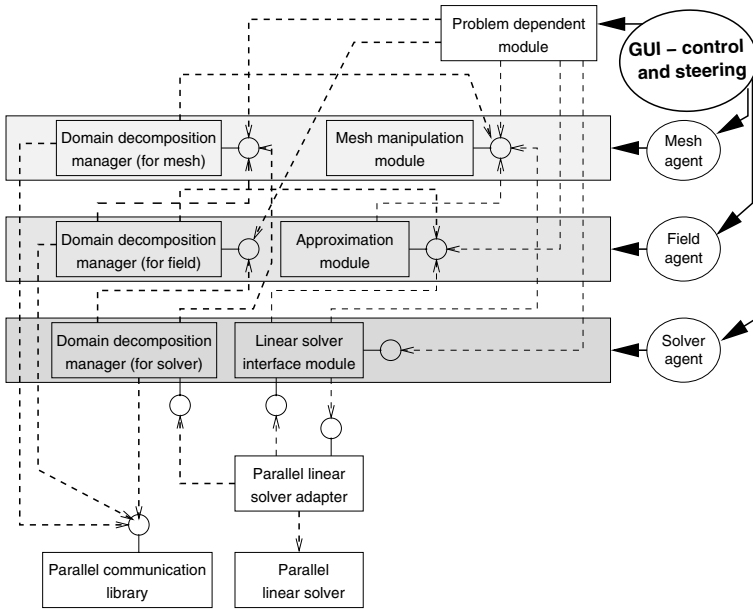


Fig. 4. The proposed agent based modular architecture for parallel finite element codes

complex issue. To resolve this issue an agent based architecture is introduced in Fig. 4. Each horizontal layer is equipped with an agent that is used to discover possible hardware-software setting for the simulation and to make decisions aiming at achieving the best performance. The information considered by agents includes data on simulation (problem, mesh, approximation, linear solver) as well as on computing system (the number of available computers/processors/cores, the speed of communication links – latency and bandwidth, current load etc.). A control-and-steering unit (possibly distributed) activates agents and provides them with user data.

7 Conclusion

The final agent architecture for computational kernels of mesh based solvers consists of several layers dealing with fundamental data structures of the code. The exchange of data and services between layers is done through fast communication links, preferably using shared memory. Each layer is composed of three components:

- the sequential part unaware of possible parallel execution
- domain decomposition manager enabling the module to communicate with other modules of the same kind using message passing and to work with them in parallel
- agent – intelligent component discovering resources and other modules, used for setting up the whole simulation environment

The proposed architecture brings additional flexibility to the process of forming hardware-software environments for performing computations. The flexibility comes with a price of much more complex management of different modules for parallel and distributed execution. The role of agents in the architecture is to provide intelligence necessary to unify all components and achieve high performance realization.

References

1. Banaś, K.: A modular design for parallel adaptive finite element computational kernels. In Bubak, M., van Albada, G., Sloot, P., Dongarra, J., eds.: *Computational Science — ICCS 2004, 4th International Conference, Kraków, Poland, June 2004, Proceedings, Part II*. Volume 3037 of *Lecture Notes in Computer Science.*, Springer (2004) 155–162
2. Szymanski, B., Varela, C., Cummings, J., Napolitano, J.: Dynamically reconfigurable scientific computing on large-scale heterogeneous grids. In Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J., eds.: *Parallel Processing and Applied Mathematics, Proceedings of Vth International Conference, PPAM 2003, Częstochowa, Poland, 2003*. Volume 3019 of *Lecture Notes in Computer Science.*, Springer (2004) 419–430
3. Kisiel-Dorohinicki, M.: Agent-based models and platforms for parallel evolutionary algorithms. In Bubak, M., van Albada, G., Sloot, P., Dongarra, J., eds.: *Computational Science — ICCS 2004, 4th International Conference, Kraków, Poland, June 2004, Proceedings, Part III*. Volume 3037 of *Lecture Notes in Computer Science.*, Springer (2004) 646–653
4. Grochowski, M., Schaefer, R., Uhruski, P.: Diffusion based scheduling in the agent-oriented computing system. In Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J., eds.: *Parallel Processing and Applied Mathematics, Proceedings of Vth International Conference, PPAM 2003, Częstochowa, Poland, 2003*. Volume 3019 of *Lecture Notes in Computer Science.*, Springer (2004) 97–104
5. Banaś, K.: On a modular architecture for finite element systems. I. Sequential codes. *Computing and Visualization in Science* **8** (2005) 35–47
6. Płażek, J., Banaś, K., Kitowski, J.: Comparison of message passing and shared memory implementations of the GMRES method on MIMD computers. *Scientific Programming* **9** (2001) 195–209
7. Banaś, K.: The application of the adaptive finite element method in large scale computations (in Polish). *Wydawnictwo Politechniki Krakowskiej, Kraków* (2004)

The Application of Agents to Parallel Mesh Refinements in Domain Decomposition Based Parallel Fully Automatic *hp* Adaptive Finite Element Codes

Maciej Paszynski

AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Cracow, Poland
paszynsk@agh.edu.pl
<http://home.agh.edu.pl/~paszynsk>

Abstract. In the *hp* adaptive Finite Element Method (FEM) applications, the computational mesh consists in finite elements with varying size h , and varying polynomial order of approximation p on finite element edges, faces and interiors. The parallel *hp* adaptive codes work on the computational domain partitioned into sub-domains with each of the sub-domains delegated to a single processor. The algorithm of parallel mesh refinements on such a distributed FE must enforce global mesh regularity rules. The paper presents the applications of multiple agents to implement the parallel mesh refinements algorithm. Agents work on distributed data structure storing FE mesh where dynamic mesh refinements are recorded by growing trees of initial mesh elements nodes. Agents located on separated sub-domains communicate in order to establish necessary actions on the distributed mesh.

1 Introduction

Parallelization of h or *hp* adaptive codes is difficult. Most implementations for distributed memory parallel computers are based on Domain Decomposition (DD) paradigm. The computational domain is partitioned into sub-domains with each of the sub-domains assigned to a single processor.

Among major undertakings to develop a general infrastructure to support DD based parallelization of PDE solvers, one has to list first of all the Sierra Environment [6] developed by Sandia National Labs, supporting h -adaptivity with polynomial order of approximation $p = 2$. The Sierra environment allows for an arbitrary domain partitioning of a current mesh but it does not support anisotropic mesh refinements. General environments to support local mesh refinements have been developed by [3], [7] and, more recently, [13]. None of them, unfortunately, turned out to be applicable directly to our *hp* codes. The only parallel *hp* codes that we are aware of, have been developed by [12], [9], [2] and [1]. None of them are *fully automatic hp* adaptive codes. The parallel fully automatic *hp* adaptive 2D and 3D FE codes were finally implemented by the group of Demkowicz [11, 10], based on the data structures described in [5].

The paper describes applications of agents to parallel mesh refinements performed during iterations of automatic *hp* adaptive 2D and 3D FE codes. Agents work on distributed data structure storing FE mesh where dynamic mesh refinements are recorded by growing trees of initial mesh elements nodes. Agents located on separated nodes communicate to enforce global mesh regularity rules which must be fulfilled during refinements on distributed mesh.

2 The Parallel Fully Automatic *hp* Adaptive Algorithm

Let us focus on the 3D Fichera model problem, presented in Figure 1, described in [10]. The problem consists in solving the Laplace equation on the 3D cube with $1/8$ smaller cube removed, see picture (a). The Neumann boundary conditions based on the known exact solution is applied on the external boundary of the cube, denoted by the red color. Zero Dirichlet boundary condition is applied on the internal boundary, denoted by the blue color. There are strong singularities in the middle point of the cube, as well as along edges of removed smaller cube. Those singularities produce large numerical error of the solution. To reduce the error, automatic *hp* adaptive strategy enforces optimal mesh refinements in those areas.

The parallel strategy starts from the coarse mesh, redistributed into processors, as it is presented in picture (b). In the example 7 processors were utilized. The coarse mesh is then globally *hp* refined. Each element from the coarse mesh is broken into 8 sons in 3D, and the order of approximation is uniformly risen by one, see picture (c). The problem is solved twice, on the coarse and on the fine mesh. For each finite element from the coarse mesh we consider various refinement strategies, and evaluate error decrease rate based on comparison of coarse and fine grid solutions. Optimal refinement is selected for each finite element from the coarse mesh. The optimal mesh after the first iteration is obtained by performing optimal mesh refinements on the coarse mesh by the parallel mesh refinements algorithm, see picture (d). The mesh must be globally consistent, which requires involving of mesh reconciliation algorithms in the parallel mesh refinements algorithm. The computational load over the optimal mesh is not well balanced, since the refinements are not uniform. There is a need to balance load, by removing some finite elements from some processors and sending them to some neighboring processors. The ZOLTAN load balancing library [14] is utilized to obtain optimal load distribution. The optimal mesh redistributed onto 7 processors becomes the coarse mesh for the second iteration, see picture (e). The mesh is again globally *hp* refined, see picture (f). The problem is again solved twice on the coarse and fine mesh, and the optimal refinements are selected for the coarse mesh, based on relative error estimations over each finite element from the coarse mesh. Finally optimal mesh refinements are executed by parallel mesh refinements algorithms, see picture (g). The iterations are performed as long as the estimated error is large. Optimal mesh obtained after 5 iterations providing solution to the Fichera problem with 1 % relative error in energy norm is presented in pictures (h) and (i).

3 Distributed Data Structure for Finite Element Mesh Supporting *hp* Adaptivity

In the data structure for distributed storage of FE mesh [5] only initial mesh elements are stored. When a finite element is broken into new son elements, refinement trees grows from initial mesh elements nodes, and newly created nodes and vertices are stored on these trees, see Figure 2. Each initial mesh element in 3D consists in 8 vertices, 12 mid-edge nodes, 6 mid-face nodes and 1 middle node. The element structure keeps pointers to all its vertices and nodes, pointers to 6 neighbors (initial mesh elements), orientation and boundary condition data. All of the vertices have polynomial order of approximation fixed to one. A finite element can be broken into 2, 4 or 8 new son elements. A face of an element is broken only if all neighboring elements are broken, including elements located on adjacent sub-domains. An edge of an element is broken only if all faces neighboring the edge are broken, including faces located on neighboring sub-domains. An element interior can be always broken. Newly created finite elements are not stored in the data structure.

To break an edge means to create 2 new mid-edge nodes and 1 new vertex and attached them as sons of the original mid-edge node. The refinement tree grows down from the mid-edge node. A face can be broken into 2 or 4 faces. To break a face into 2 new faces means to create 2 new mid-face nodes and 1 new mid-edge node. To break a face into 4 new faces means to create 4 new mid-face nodes, 4 new mid-edge nodes and 1 new vertex node. A middle node representing an element interior can be broken into 2, 4 or 8 sons. To break an element interior into 2 sons means to create 2 new middle nodes and one new mid-face node. To break an element interior into 4 sons means to create 4 new middle nodes, 4 new mid-face nodes and one new mid-edge node. To break an element into 8 sons means to create 8 new middle nodes, 12 new mid-face nodes, 6 new mid edge nodes and one new vertex node.

Each node keeps its polynomial order of approximation, type (mid-edge, mid-face or middle node), finite element degrees of freedom data, and the refinement tree data, involving: lists of sons, refinement type of the node, and a pointer to the father. Number and type of sons on the list depends on the node refinement type. Finite element method computations are performed on active elements. Their nodes and vertices are found dynamically by browsing the refinement trees from the level initial mesh elements.

4 Agent Based Algorithm for Parallel Mesh Refinements

In the parallel mesh refinement algorithm some finite elements are divided into multiple (new) elements (each rectangular element is divided into 2 or 4 sons in 2D and 2, 4 or 8 sons in 3D), new nodes are created, and new orders of approximation are set. The parallel version of the algorithm requires an enforcement of a global mesh regularity rules:

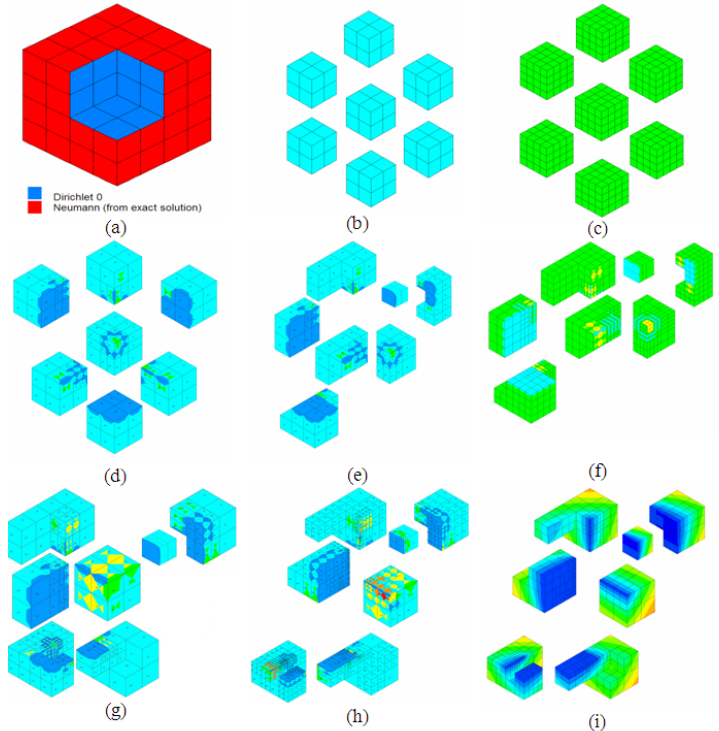


Fig. 1. Parallel fully automatic *hp* adaptive iterations

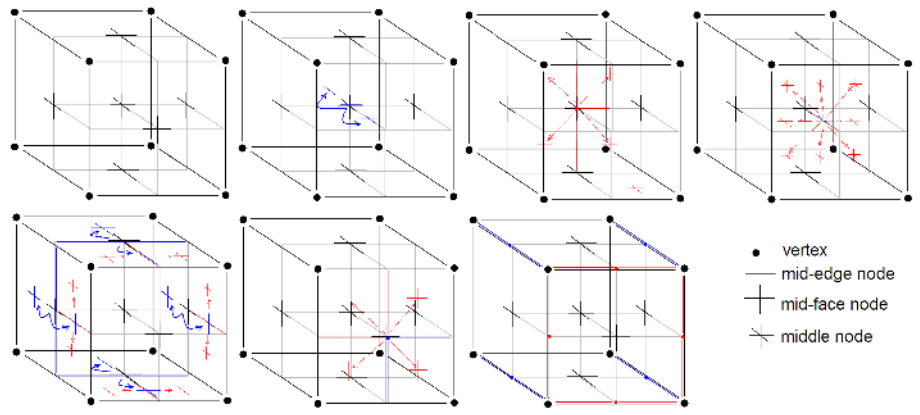


Fig. 2. Breaking an element into 2 sons, followed by breaking of one son element into 4 new son elements. The blue color denotes all refinement trees created during the first refinement, and the red color denotes refinement trees created during the second refinement. We assume that all affected element's faces and edges can be broken.

- The 1-irregularity rule: Edges of a given element can be broken only once, without breaking neighboring elements. It is necessary to check the configuration of elements adjacent on neighboring sub-domains to the element requested to be broken.
- When an interface edge is broken, then the polynomial order of newly created nodes must be established. The selection of optimal order for a node is done based on error estimations performed over the finite elements having that node. But some of the elements neighboring the interface edge are present in adjacent sub-domains. It is necessary to ask the neighboring sub-domains for the error estimation over those elements.
- The minimum rules: Order of approximation of an element interior cannot be larger than orders of approximation of the element faces, and order of approximation of a face cannot be larger than orders of approximation of faces adjacent to the edge. To enforce the minimum rule, we need to exchange orders of approximation for interface faces and edges with adjacent sub-domains.

We will discuss now the applications of agents to particular problems appearing during refinements on distributed mesh. The agents work on distributed data structure storing the mesh with refinements coded as growing trees of initial mesh elements nodes.

Enforcing 1-irregularity rule

The 1-irregularity rule requires breaking of neighboring elements if the current element must be broken for the second time. There is a need to send a request to adjacent sub-domains to break neighboring elements. The way of breaking the neighboring elements depends on the configuration of the current and the neighboring elements.

The refinement on current sub-domain is performed by an agent, that is responsible for enforcing mesh regularity rules. If the element was previously broken, the agent responsible for managing breaking the element on current sub-domain packs the information about the previous refinement of the element. The current element could be previously broken in one of many possible ways: the element could be broken into 2 sons in one of 3 possible directions, the element could be broken into 4 sons in one of 3 possible directions, or the element could be broken into 8 sons. The refinement trees have different syntax in all those cases. The agent must communicate with the agent located in other location, and request necessary refinements on the neighboring elements. The agent must pack the refinement tree and send it to agents located in adjacent sub-domains. The refinement trees are then unpacked by the agents located on adjacent sub-domains, and the decision about kind of necessary refinements of the adjacent elements are made by the remote agents by comparing received and local refinement trees.

Breaking of an element neighboring the interface

Breaking of an element involves the following steps: breaking the element interior, breaking the element faces and breaking the element edges. The element interior can be always broken. The element face can be broken only if interior of the

element neighboring current element possibly on adjacent sub-domain is broken, or the element is adjacent to the boundary of the domain. The element edge can be broken only if all faces adjacent to the edge are broken, including faces located on current and adjacent sub-domains.

Breaking of an interface face

It is necessary to ask elements located on adjacent sub-domain if its interior is broken. The agent responsible for breaking an interface face must send the question to remote agent located on adjacent sub-domain. The remote agent locates the initial mesh element containing the neighboring element on adjacent sub-domain by using the connectivity information stored in the initial mesh elements. The neighboring element is found by traveling down the refinement tree in the neighboring initial mesh element.

Breaking of an interface edge

It is necessary to locate all elements neighboring the edge located on adjacent sub-domains, and check if faces of these elements adjacent to the edge are broken. The agent responsible for breaking an interface edge must communicate with all agents located in adjacent sub-domains and request a check of state of their faces. The location of neighboring elements can be found by using the connectivity information stored in the initial mesh elements.

Setting orders of approximation for nodes located on the interface

In the first step of the fully automatic *hp* adaptive algorithm, optimal *h* refinements are performed, by breaking appropriate middle nodes, mid-face nodes and mid-edge nodes. In the next step of the algorithm, decisions about optimal orders of approximation are made. In the first version of the automatic *hp* adaptivity [4, 5, 11], decisions about optimal orders of approximations were made for edges, faces and interiors. Those decisions were made based on error estimations computed on current and neighboring elements. It was necessary to exchange information about error estimations on neighboring elements located on adjacent sub-domains. In the new version of the algorithm [8, 10], decisions about optimal order of approximation are made only for element interiors. The error estimation for the element is computed by using only local information contained in the element under consideration. The operation is fully parallel. The orders of approximation for faces and edges are computed by enforcing minimum rule.

Enforcing the minimum rule for interface edges and faces

In its first step, the minimum rule requires to adjust orders of approximation of faces as a minimum of orders of approximation of interiors of neighboring elements, possibly located on adjacent sub-domains. The agents exchange information about the orders of approximation of interiors of elements neighboring the interface between adjacent sub-domains. Then, the orders of faces located on the interface can be set by agents based on the obtained orders of approximations of elements interiors from adjacent sub-domains. The neighboring elements can be found by using connectivity information stored in the initial mesh elements. In its second step, the minimum rule requires to adjust orders of approximation

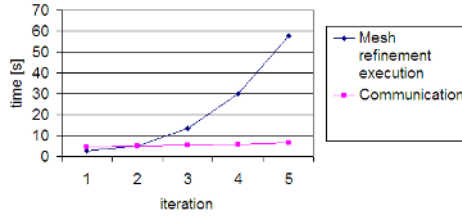


Fig. 3. Measurements of times of execution of mesh refinements versus communication

of mid-edge nodes as a minimum of orders of neighboring faces, possibly located on adjacent sub-domains. The agents exchange information about the orders of approximation of faces of elements neighboring the interface. Then, the orders of edges located on the interface can be set by agents based on the obtained orders of approximations of faces from adjacent sub-domains.

Performance test

In the parallel fully automatic *hp* adaptive codes [11], [10] there is a module responsible for execution of mesh refinements on the distributed data structure. It is like copies of identical agents, assigned one to each sub-domain, working in concurrent with the rest of the parallel code. Each agent executes on its sub-domain all mesh refinements that can be performed without communication with adjacent agents, and collects data about refinements on the interface that need to be exchanged with adjacent agents. Then, all agents exchange data according to coloring of edges of created communication graph. The algorithm is repeated unless no exchange of data with neighboring sub-domains is needed (unless the interface is not affected). It follows that the communication takes less time than execution of mesh refinements, see Figure 3. The mesh refinements execution time can be reduced by increasing number of agents working in concurrent on each sub-domain. Thus, the application of many concurrently working agents may improve the scalability of the code.

5 Conclusions

The parallel mesh refinements algorithm can be efficiently performed on the distributed data structure where the refinements are recorder by growing trees of initial mesh elements nodes. The algorithm can be realized by a set of agents working on the distributed data structure. There are two possible solutions. In the first approach each agent can be assigned to one sub-domain, and it can share its work to perform its own tasks with tasks requested by neighboring agents. This is the solution we use in our parallel codes. In the second solution, each agent can remotely invoke agents in other locations to execute its tasks on remote piece of the mesh. In such a case many agents can perform their tasks in concurrent on each sub-domain. The second solution may be more efficient, however it requires to implement synchronization mechanisms to ensure that the data structure won't be damaged by many concurrently working agents.

Acknowledgement

The support of this work under MEiN grant no. 3 T08B 055 29 is gratefully acknowledged.

References

1. Banaś K.: A Model for Parallel Adaptive Finite Element Software, Proceedings of 15th International Conference on Domain Decomposition Methods, Freie Universität Berlin, July 21-25 (2003)
2. Bastian P., Birken K., Johannsen K., Lang S., Neuss N., Rentz-Reichert H., and Wiens C.: UG - a flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, **1**, 1, (1997) 27-40
3. Das R., Hwang Y.-S., Uysal M., Saltz J., Sussman A.: Applying the CHAOS/PARTI Library to Irregular Problems in Computational Chemistry and Computational Aerodynamics. Proceedings of the Scalable Parallel Libraries Conference. Mississippi State University, Starkville, (1993) 45-46
4. Demkowicz L.: 2D *hp*-Adaptive Finite Element Package (2Dhp90) Version 2.0. TICAM Report 02-06 (2002)
5. Demkowicz L., Pardo D., Rachowicz W.: 3D *hp*-Adaptive Finite Element Package (3Dhp90) Version 2.0, The Ultimate (?) Data Structure for Three-Dimensional, Anisotropic *hp* Refinements, TICAM Report 02-24. (2002)
6. Edwards H. C.: Sierra Framework for Parallel Adaptive Multiphysics Computational Mechanics Applications. VIII US National Congress on Computational Mechanics, Austin, TX. (2005)
7. Fink S.J., Kohn S.R., Baden S.B.: Efficient run-time support for irregular block-structured applications. *J. Parallel Distrib. Comput.*, **50**, 1-2, (1998) 61-82
8. Kurtz J.: Fully Automatic *hp*-Adaptivity for Acoustic and Electromagnetic Scattering in Three Dimensions. Dissertation Proposal. The University of Texas at Austin. (2005)
9. Laszloffy A., Long J., Patra A. K.: Simple data management, scheduling and solution strategies for managing the irregularities in parallel adaptive *hp* finite element simulations. *Parallel Computing*, **26**, (2000) 1765-1788
10. Paszyński M., Demkowicz L.: Parallel Fully Automatic *hp*-Adaptive 3D Finite Element Package. ICES Report 05-33. VIIIth U.S. National Congress on Computational Mechanics, Austin, Texas. submitted to *Engineering with Computers* (2005)
11. Paszyński M., Kurtz J., Demkowicz L.: Parallel Fully Automatic *hp*-Adaptive 2D Finite Element Package. ICES Report 04-07. *Computer Methods in Applied Mechanics and Engineering*, **195**, 7-8, (2006) 711-741
12. Remacle J. F., Xiangrong Li, Shephard M. S., Flaherty J. E.: Anisotropic Adaptive Simulations of Transient Flows using Discontinuous Galerkin Methods. *Int. J. Numer. Meth. Engng.* **62**, 7, (2005) 899-923
13. Yelick K. et al.: Titanium: A High-Performance Java Dialect. ACM 1998 Workshop on Java for High-Performance Network Computing, Stanford, California, February (1998)
14. ZOLTAN: Data Management Services for Parallel Applications, <http://www.cs.sandia.gov/zoltan>

Multiagent Simulation of Physical Phenomena by Means of Aspect Programming

Sławomir Bieniasz, Stanisław Ciszewski, and Bartłomiej Śnieżyński

AGH University of Science and Technology, Department of Computer Science,
Krakow, Poland

bieniasz@agh.edu.pl, scisz@iisg.agh.edu.pl, sniezyn@agh.edu.pl

Abstract. Along with the evolution of the numerical methods new software methodologies have been developed. Goal of our research is to apply Aspect Oriented Programming (AOP) in the development of multiagent simulation system. In this paper theoretical model of aspect-multiagent system is presented, its architecture and implementation is described. Results of experiments performed conclude the work. The model considered here can serve as a design tool for foundry processes, especially to design conditions for cooling of a casting leading to desired crystal structure.

1 Introduction

Simulation software is usually build using structural programming techniques, which resembles functions and equations found in an application domain. This leads to a clear structure of a model which is typically specialized in a narrow field of the application.

The next milestone of software engineering, object orientation paradigm, has been introduced to numerical methods last years. The advantages of this approach are well known, but on the field of simulation of natural phenomena one important drawback have to be pointed. The data centric encapsulation, which perfectly plays its role on business model field, is hard to be properly done when common state is more shared then divided.

The paper presents the next step approach to this problem i.e. composition of Agent Methodologies and Aspect Oriented Programming (AOP) [1]. In such an approach the state of the system is modelled as an environment, typically a mesh, which could be observed and changed by set of agents. The physical phenomena are represented as their societies, mutual relations and specialized roles. The separation of crosscutting concerns is used to distinguish both the physical phenomena and parts of the software.

The paper develops those ideas as follows. The second section gives the description of some related works on the field of aspect-multiagent approach. The third section presents a theoretical model of such system which is centered upon the separation of physical phenomena concept. The demonstration system Aspect MAFES is described in the forth section. The real life examples are shown

in the fifth section. The model considered there can serve as a design tool for foundry processes, especially to design conditions for cooling of a casting leading to desired crystal structure.

2 Related Work

Aspect programming is used to build multiagent systems for several years. Kendal [2] presents how aspects can be used to model, design and implement agent roles, which focuses on the position and responsibilities within an overall multiagent system. Examples of roles are broker, mediator, and bureaucracy.

Garcia et al. [3] describe using aspect-oriented programming to separate concerns in multiagent systems development. Two main groups of aspects are proposed: agency aspects (such as autonomy, collaboration, and mobility), and generic aspects (e.g. exception handling, and persistence). Additionally, application of computational reflection is proposed to manipulate separately interagent level properties.

Robbes et al. [4] propose application of aspect-oriented programming in multiagent systems that are based on the Aalaadin [5] MAS model. It is done on two levels: conceptual (where aspect is unified with the group concept) and implementation. Aspects that are proposed are connected with messaging (message building, messaging strategy, distributed messaging), agent lookup, visualization and debugging, and flow control.

In this work an approach similar to the Garcia's and Robbes' is used in the numerical simulation domain.

3 Theoretical Models

Let us denote by S a set of states of universe, by P a set of observations, by A a set of actions, and by Ag a set of agents. Classically, agent is determined as pair of methods [6]:

$$a_i = (see_i, action_i), \quad (1)$$

where:

$$see_i : S \rightarrow P, \quad (2)$$

$$action_i : P^* \rightarrow A, \quad (3)$$

where P^* is a set of sequences of observations from P . Function see_i receives information about the environment and transforms it into some internal observation. The $action_i$ chooses an action that should be executed according to the history of information received.

In this work an environment is treated as a deterministic state transition function:

$$env : A \times S \rightarrow S \quad (4)$$

With such assumption a consecutive application of see_i , $action_i$ and env is valid, only if observations and actions that are executed are alternate. To overcome this problem the additional function:

$$\sigma : Ag \rightarrow \mathbb{N}, \tag{5}$$

is introduced to assign a priority to every agent. Priorities should be assigned in a way that agent actions and observations with the same priority are alternate.

To handle more sophisticated functionality, the basic structure of the agent system is extended in the following way:

- the state of the universe is modelled as a family of partial functions,
- to introduce slices, instructions used to code see_i and act_i are considered with static data flow dependencies extended to the universe,
- finally, physical representation of aspect waving is defined.

Let $\Gamma \neq \emptyset$ be a set of entities (or attributes) in the universe. For any $\gamma \in \Gamma$ we define the set of its values V_γ . By a $\mathcal{V} = \bigcup_{\gamma \in \Gamma} V_\gamma$ the total space of values is noted. With this annotations the function:

$$s : \Gamma \ni \gamma \rightarrow s(\gamma) \in V_\gamma \subset \mathcal{V} \tag{6}$$

denotes the state of the universe in a given time. We assume that for all $\gamma \in \Gamma$ the set of values V_γ contains a special *none* element. For a given state the statement $s(\gamma) = none$ means that γ cannot be taken into account in the state s .

The internal structure of an agent is a pair (\mathbb{I}, \ll) , where \mathbb{I} denotes the set of instructions, and \ll is a static data flow relation. It is assumed that \ll is an order, as usual. This relation is naturally extended to the entities of Γ by instruction value dependencies.

Definition 1. *The slice in \mathbb{I} is a pair $A = (A^\Gamma, A^\mathbb{I})$ where $A^\Gamma \subset \Gamma$ and $A^\mathbb{I}$ is set of instructions satisfying following conditions:*

- if $i \in A^\mathbb{I}$ then $\forall a \in \Gamma$ such that: $a \ll i$ then $a \in A^\Gamma$,
- if $i \in A^\mathbb{I}$ and $a \in A^\Gamma$ then $\forall j \in \mathbb{I}$ such that: $a \ll j \ll i$ then $j \in A^\mathbb{I}$.

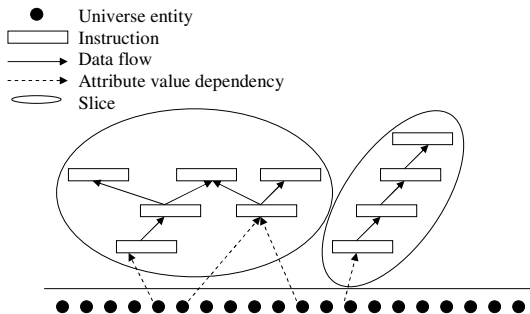


Fig. 1. Slices in the MAS

The definition is illustrated in Fig. 1. The dots below depicts the elements of the universe. The instructions, marked as rectangles, depend on their values. The data flow relation between instructions transfer the value dependency to the further instructions. The two slices are marked by ovals.

Let us remark, that any slice A could be treated as a sum of intersection with agent methods. It is denoted as: $A_{see} = A \cap see$ and $A_{action} = A \cap action$.

Definition 2. *Two slices A and B are independent iff $A^\Gamma \cap B^\Gamma = \emptyset$ and $A^\mathbb{I} \cap B^\mathbb{I} = \emptyset$.*

Our example (Fig. 1) shows such a situation.

Definition 3. *A slice A is complete iff A and the smallest slice which contain its complement $(\Gamma - A^\Gamma, \mathbb{I} - A^\mathbb{I})$ are independent.*

The completeness of slices A should be understood, in the context of this article, as a separation of physical phenomena. Each of them could be distinguished by a set of entities and their values, which constitutes restriction of universe state to $A^\Gamma \subset \Gamma$.

Definition 4. *A slice A separates phenomenon iff there exists a complete slice B such that: $A^\Gamma = B^\Gamma$ and $A^\mathbb{I} \subset B^\mathbb{I}$*

Let us consider a slice A and a family of $\{X_i\}_{i=1,2,\dots,k}$ such that: $X_i = (X_i^\Gamma, X_i^\mathbb{I})$, where: $X_i^\Gamma \subset \Gamma$ and $X_i^\mathbb{I} \subset \mathbb{I}$ for $i = 1, 2, \dots, k$.

Definition 5. *A slice A is properly extended (weaved) by a X_i when $A + X_i = (A^\Gamma \cup X_i^\Gamma, A^\mathbb{I} \cup X_i^\mathbb{I})$ is also a slice.*

Definition 6. *With the above notation the aspect system separates phenomena if all partial proper extensions for set $U \subset \{1, 2, \dots, k\}$: $A + \sum_U X_i$ separates phenomena; where $A + \sum_U X_i$ means $(\dots((A + X_{i_1}) + X_{i_2})\dots + X_{i_n})$ for $i_j \in U$, $j \in \{1, 2, \dots, n\}$, $n = |U|$.*

4 System Description

During refactoring of the MAFES system (Multi Agent Finite Environment System) [7] several aspects were identified. As a result an Aspect MAFES system was built according to a theoretical model described above.

System consists of a multiagent basic model, with which separate phenomena, control, and visualization aspects are weaved.

Multiagent basic model consist of an environment in the form of the node matrix, and a set of agents operating on these nodes. The environment is used to model a space. Every node has references to the closest neighbors. Agents observe the universe state, and affect the part of it, by changing selected entities values, which are in theirs activity range.

Aspects are used to assign tasks to agents. Task assignment is connected with adding appropriate functionality to agents to perform given task and weaving

appropriate resources and attributes to the universe. Aspects are used to add physical phenomenons (such as motion, heat exchange, crystallization, etc.) and other activities (control, visualization, storing simulation results, etc.).

The following physical phenomenon aspects are implemented in the system: heat exchange, crystallization, and motion. Schemas of these aspects are presented in Fig. 2.

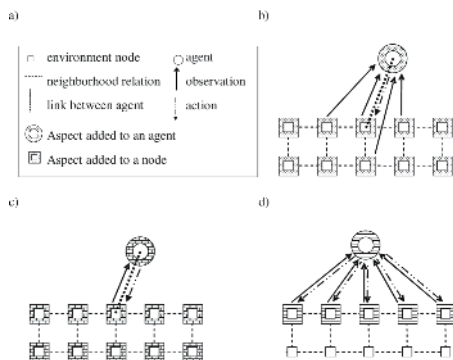


Fig. 2. Aspects of phenomena: a) legend, b) aspect of heat exchange, c) aspect of crystallization, d) aspect of motion

Aspect of heat exchange (Fig. 2-b) adds to the environment nodes the following properties: temperature, specific heat, heat conductivity, and density. To the agent activity temperature change algorithm is added. According to the theoretical model, it is build of two parts: observation that gets temperature and other parameters of the linked node and its neighbors, and action that estimates differentials of the temperature and changes the temperature of the linked node.

Aspect of crystallization (Fig. 2-c) is implemented using model of eutectic equilibrium. It adds to the environments attributes representing all necessary crystallization parameters (FS, N, R, CGN). Agent with this aspect weaved observes temperature and these parameters. Using actual and historical observations (history is necessary to recognize undercooling point), it estimates temperature change and applies it to the linked node.

Aspect of motion (Fig. 2-d) causes that agent observes and affects all moving space (nodes that represent it). Such an agent transfers properties of these nodes according to the speed of motion. This action should be done before these mentioned above. As a result, agents with this aspect weaved have higher priority.

Three more aspects are used in Aspect MAFES: control (e.g. used to control the cooling), visualization (to present the state of simulation) and result storing (to store all necessary simulation parameters). Because of the lack of space, they are not described here.

Aspect MAFES is implemented using AspectJ [8, 9] language, a simple, uniform and very practical extension of Java. This language adds several tools that provide a support to modular implementation of crosscutting concerns.

Aspects extend an environment using introductions to nodes. Agents functionality is extended by defining cut points and advices.

5 Experiments

To test the Aspect MAFES system, several experiments were done. Results of two of them are presented below.

5.1 First Experiment: Heat Exchange and Motion

Two phenomena are considered in this experiment: heat exchange and a motion. A temperature map for a sequence of time stamps is presented in Fig. 3-(I). Thermophysical parameters for layers are presented in Table 1. They are chosen in a way that heat exchange in the upper layer is much slower then in the lower layer.

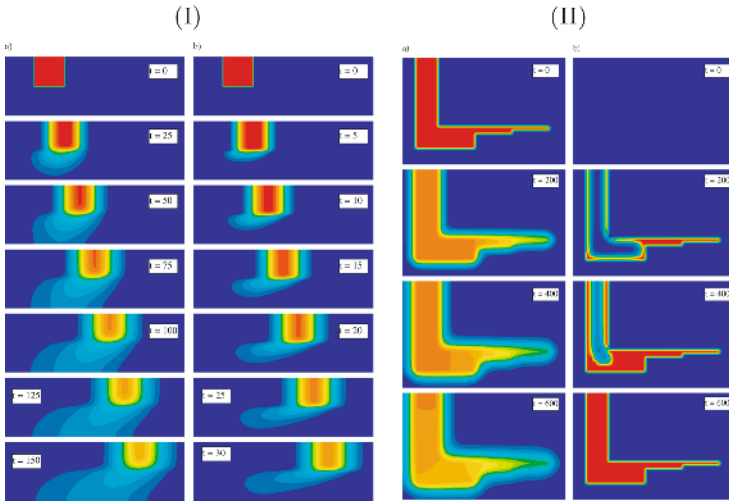


Fig. 3. (I) – temperature map for consequent time stamps in the first experiment – in the right column upper layer is moving five times faster then in the left column; (II) – temperature (left column) and FS coefficient (right column) maps for consequent time stamps in the second experiment

Table 1. Termophysical parameters used in experiments

Parameter	First Experiment		Second Experiment	
	Lower layer	Upper layer	Cast	Mould
Heat conductivity [W/K b]	20	5	210	1
Specific heat [J / (kg K)]	300	1180	1180	1333
Density [kg / m3]	1200	2250	2550	1500
Latent heat [J / kg]	-	-	3.73E+5	-
Eutectic temperature [°C]	-	-	610	-

Two speeds of the upper layer v_u were tested. Left column (a) is a result for the speed $v_u = 0.002[m/s]$, right (b) is for $v_u = 0.01[m/s]$. Time stamps are chosen in a way that pictures in the same row have the same positions of the upper layer.

5.2 Second Experiment: Heat Exchange and Crystallization

In this experiment heat exchange and crystallization in a foundry process model are considered. Cast has a stepped shape. Results of simulation are presented in Fig. 3-(II). Rows represent consecutive time stamps. In the left column (a) temperature map is presented, in the right column (b), FS coefficient, which represent the fraction of the solid phase, is shown. Thermophysical parameters for mould and cast are presented in Table 1. Initially, temperature of the cast is 700 [°C], and of the mould is 20 [°C].

The shape of a cast causes a big difference in the speed of cooling down of specific regions. As a consequence, microcrystalline structure is non-uniform.

6 Conclusions

Presented concept of the aspect-multiagent simulation system may be applied to a wide variety of practical problems, which must take into consideration cooperation of superimposed processes occurring within the common environment.

Some experiments show usefulness of the proposed computational approach. Further development will include improvement of system versatility through incorporation of other agent and aspect types as well as adaptation to other applications.

There are several main advantages resulting from AOP application during development of the simulation system:

- there is a straight correspondence between physical phenomenon (used during analysis) and aspect (used in software development);
- code for the every physical phenomenon and other concerns (such as control, visualization, storing results) is modularized (written in one unit of compilation);
- there is an analogy between phenomenon overlapping and aspect weaving;
- building versions of the system for specific requirements is very simple (it is enough to weave appropriate aspects);
- implementation of configuration of the system is easier;

As a result, structure of the code is much simpler and easy to understand and maintain.

References

1. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwing, J.: Aspect-oriented programming. In: Proc. of ECOOP'97, Springer Verlag (1997) 220–242
2. Kendall, E.: Aspect-oriented programming for role models. In: In Proc. of The Aspect-Oriented Programming Workshop at ECOOP'99. (1999)

3. Garcia, A., Chavez, C., Silva, O., Silva, V., Lucena, C.: Promoting advanced separation of concerns in intra-agent and inter-agent software engineering. In: Workshop on Advanced Separation of Concerns in Object-oriented Systems (ASoC) at OOP-SLA'2001, Tampa Bay, Florida, USA (2001)
4. Robbes, R., Bouraqadi, N., Stinckwich, S.: An aspect-based multi-agent system. In: Research Track of the ESUG 2004 Smalltalk Conference, Kthen (Anhalt), Germany (2004)
5. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: Third International Conference on Multi-Agent Systems (ICSMAS98). (1998) 128–135
6. Weiss, G., ed.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, London (1999)
7. Bieniasz, S., Cetnarowicz, K., Nawarecki, E., Kluska-Nawarecka, S.: Agent-based simulation in finite element environment. In Binder, Z., ed.: Management and Control of Production and Logistics: a proceedings volume from the 2nd IFAC/IFIP/IEEE conference. Volume 1., Grenoble, Oxford, Pergamon (2001) 545–550
8. AspectJ Team: The AspectJ documentation. (<http://www.eclipse.org/aspectj/>)
9. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.: An overview of AspectJ. In: Proc. of the ECOOP 2001. Volume 2072 of Lecture Notes in Computer Science., Springer (2001)

Modelling Tactical Driving Manoeuvres with GA-INTACT

H. Tawfik¹ and P. Liatsis²

¹ School of Computing, Liverpool Hope University, Liverpool, UK
tawfikh@hope.ac.uk

² School of Engineering and Mathematical Sciences, City University, UK
p.liatsis@city.ac.uk

Abstract. This work concerns the design and development of a driving simulation system, which exhibits intelligent driving behaviour at the tactical level, as part of a traffic simulation environment. Our tactical driving system using genetic algorithms, named GA-INTACT, accounts for the subject vehicle and other vehicles positions and speed parameters in the surrounding traffic condition, and selects favourable speed change and lane transition actions for the 'subject' vehicle, according to safety, speed and driving behaviour criteria. Simulation results demonstrated that the adoption of the Genetic Algorithms approach for obtaining near-optimum driving solutions eliminates the need for learning driving patterns, and allows the efficient handling of the complex nature of tactical driving modelling problem. The role of the driving behaviour in influencing the outcome of the driver's decision is emphasised, an aspect that was not treated sufficiently in previous tactical driving simulation approaches.

1 Introduction

The development of Autonomous Vehicle Navigation (AVN) systems remains an active research area for the automotive industry, and several AVN systems [3,13] have been tested in structured environments, such as motorways. A key issue to the success of any AVN system is the systematic performance evaluation of the underlying sensing algorithms [5] in a variety of traffic scenes and driving scenarios.

The Road Simulation Environment (ROSE) [12] has been developed as a traffic and driving simulation platform for testing obstacle detection, lane detection and driving modelling systems. This has been achieved through the construction of a visual database of traffic scenes; and the generation of geometrically valid 3-D roads [7], and driving scenarios by simulating driving decision-making procedures.

We focus here on the tactical driving part which contributes to limited research in the modelling of this complex driving activity. Our tactical driving system, GA-INTACT, selects favourable tactical driving actions for the 'subject' vehicle according to safety, speed and driving behaviour criteria, using a genetic algorithm.

2 Related Tactical Driving Simulation Research

Considerable progress was made in automating various strategic and operational driving aspects, compared to limited success in developing intelligent tactical driving

models [4]. Typical applications of tactical modelling targeted automatic cars and robots driving in simulated environments and adopted various rule-based, neural networks and evolutionary techniques. The main challenges associated with rule-based approaches [4,10] are the difficulty of modifying existing rules and implementing new ones, and poor results in unanticipated driving situations. Typical problems that arise with neural networks solutions [1,9] include a biased behaviour towards a particular driving style, exhibiting similar mistakes to those in the training data, and the lack of a criterion for the determination of satisfactory a representation for tactical driving in the training set. Other research attempts used genetic algorithms (GAs) to model tactical driving tasks [2,11] by exploiting their power in solving complex problems without the need for a comprehensive understanding of tactical driving as for the rule-based case or for a representative training set as with neural networks.

3 Intelligent TACTical Driving System Based on Genetic Algorithms (GA-INTACT)

The objective of the driving simulation module of ROSE is to provide driving scenarios where vehicles drive at desirable speeds, maintain safety gaps, and closely resemble real driving, in terms of speed changes and lane transitions.

A ‘Subject driver’ centred approach was adopted such that driving journey refers to the changing driving situation in the immediate vicinity of the subject vehicle driver. The subject vehicle driving actions in terms of lane and speed changes are determined by GA-INTACT. A dual carriageway road type with two lanes on each traffic direction (Fig.1) was chosen for this application. It permits a variety of manoeuvres to be performed, and is generalisable to other road structures such as Motorways.

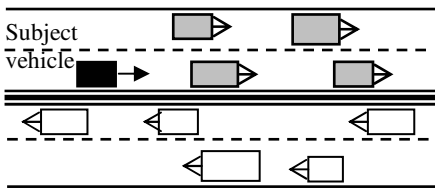


Fig. 1. Layout of the dual carriageway



Fig. 2. GA solution representation

A number of application specific assumptions were made, as follows:

- Only the subject vehicle has the ability to perform intelligent tactical manoeuvres. The driving scenarios of other vehicles are predetermined and prior information with regard to each vehicle’s position and speed, at any stage, is also available.
- A flowing traffic without interruptions, roundabouts or junctions, is assumed.
- A five-second interval (FSI) is the updating period for the tactical driving steps as it allows adequate time to perform realistic speed changes and lane transitions [12].
- The two-second minimum gap: This safety distance is the distance required to allow for a two second time gap to the vehicle ahead, at the subject vehicle speed, such that:

$$D_{\text{two_sec}} = V \times 2 \tag{1}$$

where $D_{\text{two_sec}}$ is the two-sec distance in metres, and V is the vehicle speed in (m/s).

- Driver Behaviour Model (DBM): The DBM input takes the parameters of the driver’s age, gender, experience, and personality, and returns a 0 to 1 output, corresponding to a range of very risky to very cautious driving behaviours, as follows:

$$DBM = w_1 DBM_A + w_2 DBM_E + w_3 DBM_{P/G} \tag{2}$$

where DBM_A , DBM_E , and $DBM_{P/G}$ are the different DBM contributions resulting from the driver’s age, experience and personality/gender factors, respectively [12].

3.1 Genetic Algorithms

A genetic algorithm search was adopted for the selection of optimum tactical manoeuvres for the subject vehicle as it can provide a powerful means of obtaining optimum solutions for complex problems and eliminate the need for obtaining training data. GAs are stochastic search methods that adopt mechanisms of natural selection and genetics for carrying out their optimisation tasks [6].

The GA solution in GA-INTACT is a tactical decision in the form of a change in speed, acceleration rate, and lane and the type of lane transition, which is then optimised with respect to tactical driving criteria. We use a 14-bit binary string that encodes the speed, acceleration, lane parameters (Fig. 2), as follows:

- V_{Subject} is the target speed of the subject vehicle at the end of the FSI. The 7 bits allow for a 1-kph resolution of a 0 to 128 kph speed range.
- ACS (Acceleration Scale): this scaling factor corresponds to the change in acceleration/deceleration over the FSI [8]. A 3-bit field allows for scenarios such as (000) for a ‘very sharp’ acceleration rate and (111) for a ‘very smooth’ deceleration rate.
- LN is the subject vehicle’s lane number at the end of the FSI where LN = 0 is for the slow lane and LN = 1 is for the fast lane.
- LTS (Lane Transition Scale): this reflects the manner by which the lane change is performed in terms of smoothness/sharpness. LTS ranges from ‘very smooth’ fast-to-slow lane transition (LTS = 7/7) to ‘very sharp’ slow-to-fast transition (LTS = 0/7).

3.2 GA Fitness Function Design

Four main design optimisation criteria were defined, namely safety, speed, driving behaviour, and ‘keeping to the slow lane’. The overall GA fitness function can be described as the weighted sum of the 4 evaluation measures, as follows:

$$GA \text{ fitness} = w_1 \cdot f_{\text{Safety}} + w_2 \cdot f_{\text{Speed}} + w_3 \cdot f_{\text{Behaviour}} + w_4 \cdot f_{\text{Slow}} \tag{3}$$

Three conditions were imposed to reject unrealistic or unlawful driving solutions at every FSI manoeuvre, based on min. gap and max. speed and acceleration limits:

- $G(\cdot) \geq G_{\min}(\cdot)$: Gap to the vehicle ahead which is below minimum safety gap.
- $V(\cdot) \geq V_{\text{Opt}}(\cdot)$: Subject vehicle drives faster than its optimum speed V_{Opt} .
- $a(\cdot) \geq a_{\text{Max}}(\cdot)$: Vehicle exceeds its max. acceleration/deceleration rate a_{max} [8].

3.2.1 Safety Fitness (f_{Safety})

The objective of the safety function is to drive the subject vehicle such that the gap to the vehicle ahead is reduced, while maintaining the minimum safety gap, as follows:

$$f_{\text{Safety}} = \frac{S_{\text{Safety}}}{S_{\text{Safety}} + (G(t+T) - G_{\min}(t+T))} \tag{4}$$

where: $T = \text{FSI}$ (i.e., 5 seconds),

$G(t+T)$ is the actual gap ahead of the subject vehicle at the end of the manoeuvre.

$G_{\min}(t+T)$ is the minimum gap ahead of the subject vehicle at time $(t+T)$.

The G gap is the distance from the subject vehicle to the vehicle ahead.

G_{\min} is based on the two-second rule, with the addition of the DBM related term [12].

S_{Safety} is a scaling parameter based on the initial Gap $G(t)$ is and minimum gap $G_{\min}(t)$.

3.2.2 Speed Fitness (f_{Speed})

The f_{Speed} function encourages solutions according to which the subject vehicle drives at speeds near or equal to the optimum speed, as follows:

$$f_{\text{Speed}} = \frac{S_{\text{Speed}}}{S_{\text{Speed}} + (V_{\text{Opt}}(t+T) - V_{\text{Subject}}(t+T))} \tag{5}$$

The optimum speed V_{Opt} is dependent on the road design speed $V_D(\cdot)$, the vehicle's acceleration capabilities, and its driver behaviour DBM [12].

$V_{\text{Subject}}(t+T)$ is the subject vehicle speed at the end of the FSI.

$V_{\text{Opt}}(t+T)$ is the optimum subject vehicle speed at the end of the FSI

S_{Speed} is the f_{Speed} function scaling parameter based on $V_{\text{Subject}}(t)$ is the initial subject vehicle speed and $V_{\text{Opt}}(t)$ is the initial optimum subject vehicle speed.

3.2.2 Driving Behaviour Fitness ($f_{\text{Behaviour}}$)

The $f_{\text{Behaviour}}$ gives reflects the compatibility of the subject vehicle manoeuvre with the driving behaviour (DBM). It encourages tactical actions in the form of LTS, ASC, and SCS (Speed Change Scale) [12], which are coherent with DBM and discourages actions that are inconsistent with it. $f_{\text{Behaviour}}$ can be written as:

$$f_{\text{Behaviour}} = f(\text{LTS}, \text{ACS}, \text{SCS}, \text{DBM}) \tag{6}$$

DBM is the subject driver behaviour model output value, LTS is the lane transition scale, ACS is the acceleration rate scale, and SCS is the speed change scale [12].

A heuristic model was formulated with data points that define the correspondence between the DBM, LTS, SCS and ACS input terms and the output value of $f_{\text{Behaviour}}$ [12]. According to this model, when $\text{DBM}=0.9$ (overcautious drivers), high performance

scenarios ($f_{\text{Behaviour}} > 0.8$) involve low acceleration rates, decreasing speeds ($\text{SCS} \setminus \text{ACS} > 0.6$), and smooth lane transitions from fast to slow lanes ($\text{LTS} > 0.7$). On the other hand, very low fitness values are produced, when a ‘cautious’ driver carries out high acceleration and speed rates and sharp lane transition manoeuvres.

3.2.4 ‘Keeping to Slow Lane’ Fitness ($f_{\text{Slow lane}}$)

This fitness term comes to noticeable effect, when there are ‘almost’ equal gaps ahead of the subject vehicle to the immediate vehicles on both slow and fast lanes, with these vehicles driving at similar speeds. The influence of $f_{\text{Slow lane}}$ is such that, when a similar traffic condition are present at both lanes, the subject vehicle would end up on the slow lane. The $f_{\text{Slow Lane}}$ function was formulated as follows:

$$f_{\text{Slow lane}} = (\text{LN}_{\text{max}} - \text{LN}) \times \left(\frac{S_{\text{Slow lane}}}{S_{\text{Slow lane}} + |V_{\text{Slow}} - V_{\text{Fast}}| + |G_{\text{Slow}} - G_{\text{Fast}}|} \right) \quad (8)$$

where V_{Slow} and V_{Fast} are the speeds of the vehicles ahead, on slow and fast lanes.

G_{Slow} and G_{Fast} are the gaps of the vehicles ahead on the slow and fast lanes, respectively. If there are no vehicles ahead, then $G_{\text{Slow/Fast}} = 200$ m, and $V_{\text{Slow/Fast}} = 128$ kph.

LN is the lane number of the subject vehicle at the end of the FSI.

LN_{max} the maximum number of lanes per traffic direction number.

The $S_{\text{Slow lane}}$ scaling parameter adjusts the sharpness of $f_{\text{Slow lane}}$ based on DBM [12].

4 Simulation Results

This section presents examples of tactical driving results produced by GA-INTACT. A 3 FSI traffic journey length was selected to evaluate the performance of the system.

A graphical representation of the behaviour of the subject vehicle’s actions is presented with plots of the key GA optimisation criteria as follows:

Subject vehicle speed (V_{Subject}) and optimum speed (V_{Opt}) Vs (FSI) or (T);

The minimum gap (G_{min}) and actual gap to the vehicle ahead (G) Vs time steps (T).

Lane number (LN), LTS and ACS values at each time step (T).

The speed and distance values are approximated to the nearest kph for speed calculations, and nearest meter for gap calculations, while LTS and ACS values are approximated to the nearest decimal. An initial LTS and ACS value of 0.5 is assumed.

Due to the dynamic nature of the driving conditions and constraints, tactical manoeuvres are assessed in comparison to the optimum speed and minimum safety gaps.

Our traffic scenario has the following conditions: Design speed=120 kph, Subject vehicle starting lane LN=0, Starting speed =80 kph, speed of the first vehicle ahead on the slow lane=90 kph, and speed of the first vehicle ahead on the fast lane=100 kph.

In order to observe the effect of the driving behaviour on the tactical manoeuvres, the same traffic conditions were simulated with three DBM values: DBM=0.9 ‘over-cautious’ driver, DBM=0.5 ‘standard’ and DBM=0.1 ‘risky’ driver.

We use the U.K road layout standards in which the slow lane refers to the left lane while the fast lane refers to the right lane. This layout structure and subsequent calculations can be changed in the simulation to meet other road layout conventions.

Case 1: overcautious driver (DBM =0.9), Figs 3&4.

1st manoeuvre (Time step 0 to 1): the subject vehicle accelerates from 80 kph to 102 kph to match its optimum speed (V_{Opt}), thus reducing the gap ahead from 60m to 58m, in a ‘smooth’ manner (ACS=0.4), on the same lane.

2nd manoeuvre (Time step 1 to 2): the gap ahead of the subject vehicle is reduced to the optimum gap from 58m to 50m on the same lane, by decelerating smoothly (ACS = 0.7) to a speed of 92 kph, a speed level near that of the vehicle ahead.

3rd manoeuvre (Time step 2 to 3): The optimum gap of 48m is matched, by re-adjusting the subject vehicle’s speed to that of the vehicle ahead on the left lane. The speed is reduced smoothly (ACS=0.7) from 92 kph to 90 kph.

Case 2: standard driver (DBM =0.5), [12]

1st manoeuvre: the subject vehicle accelerates from 80 kph to match its optimum speed of 104 kph, thus reducing the gap ahead from 60m to 57m, close to the optimum gap. The subject vehicle increases speed in a ‘smooth’ manner (ACS=0.4).

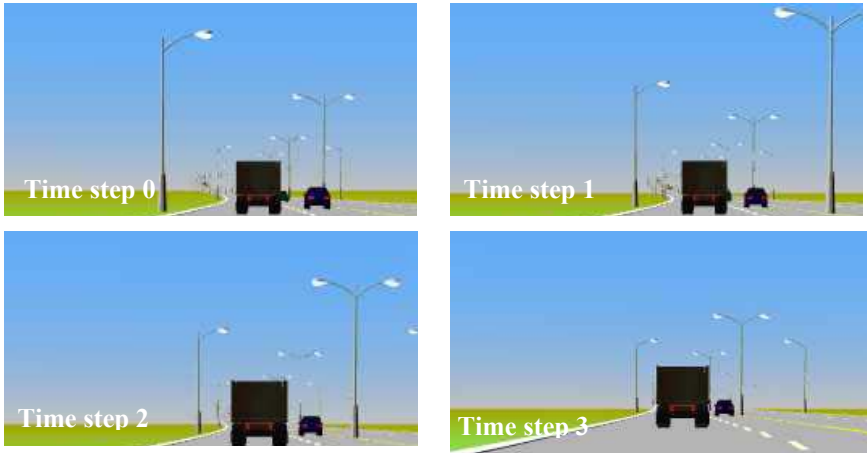


Fig. 3. Three 5-second step journey from the subject vehicle’s perspective for case 1

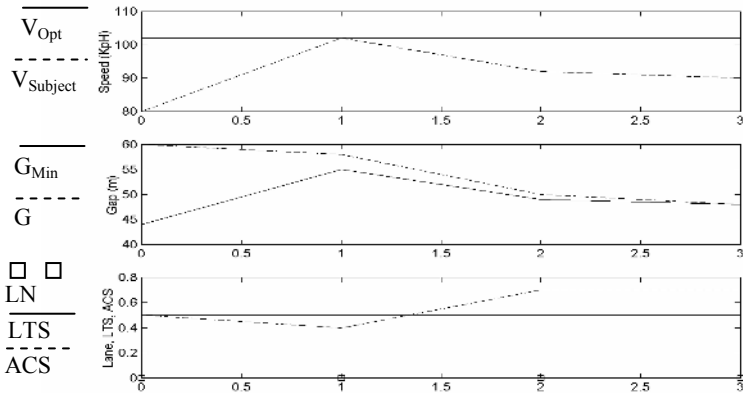


Fig. 4. Five-second time step changes of the subject vehicle’s behaviour, for case 1

2nd manoeuvre: the gap ahead of the subject vehicle is decreased to 52m and is therefore kept near the optimum gap of 50m. This is achieved by decelerating smoothly ($ACS = 0.6$) to a speed of 91 kph, and making no lane change.

3rd manoeuvre: The subject vehicle carries out a smooth lane transition ($LTS=0.4$) from the slow to the fast lane ($LN=1$). The speed is increased from 91 kph to the optimum of 104 kph on the fast lane, in a smooth manner ($ACS=0.4$). The gap to the 'new' fast-lane vehicle ahead, which had been traveling at 00 Kph, is now very large.

Case 3: risky driver ($DBM = 0.1$), [12]

1st manoeuvre: the subject vehicle accelerates from 80 kph to 107kph, a speed close to its optimum 111 kph. This speed allows a 56m gap to the vehicle ahead which is a near-optimum gap length, as the minimum gap is 55m. The vehicle also stays on the slow lane, and carries out the speed increase in a 'very sharp' manner ($ACS=0.1$).

2nd manoeuvre: The subject vehicle carries out a very sharp lane change ($LTS=0.1$) from the slow to the fast lane ($LN=1$). This allows it to increase its speed from 107 kph to the optimum 111 kph s. A sharp acceleration type was carried out ($ACS=0.3$).

3rd manoeuvre: no speed or lane change, reducing the gap is reduced from 90 to 80m.

Results show that GA-INTACT aims at driving at maximum possible speeds and maintaining minimum gaps. When there exists a wide range of optimisable speed and safety gap combinations, high DBM 'cautious' drivers, have the tendency of traveling on the slow lane and optimising the safety distance. 'Standard' drivers produce a driving action that follows the general need to optimise the speed and the safety distance, changing lanes when necessary. 'Risky' drivers increase their speed and travel on the fast lane. In addition, varying DBMs result in different approaches to speed and lane changes. Cautious drivers perform smooth, acceleration and lane transitions, while risky drivers perform these manoeuvres in a sharp manner.

Optimisation of the safety and speed criteria can be thought of as driving at the maximum possible speed, while keeping the safety distance to the other vehicles.

The use of DBM represents a distinct property of the GA approach in our GA-INTACT system, and allows the generation of more realistic and broader types of tactical manoeuvres, and the possibility to 'steer' the subject vehicle's driving journey.

The current 14-bit GA solution size means that the problem under consideration has a relatively small search space. The GA efficiency would be more apparent when using GA-INTACT for more complicated configurations. For more than 2 lanes per direction, the chromosome size needs to be increased such that it allows enough resolution to accommodate for extra LTS patterns. In a more sophisticated traffic scenario in which other vehicle are equipped with GA-INTACT, the solution size can increase significantly to accommodate for interactions between vehicles driving in the vicinity of each other. This leads to a larger search space to explore and would require more parallel optimisation of the tactical manoeuvres of all interacting vehicles.

5 Conclusions

The GA-INTACT system adopts a genetic search approach to generate high-performance tactical driving solutions with regards to safety, speed and driving behaviour criteria. The driver behaviour element was particularly emphasised during the

tactical driving design. GA-INTACT performs a variety of speed and lane change actions at various traffic situations, and generates tactical manoeuvres which optimise driving safety, efficiency and the driver's behaviour-manoevre compatibility criteria.

The work may be extended to a more versatile driving simulation system, which accommodates for driving decisions that relate to issues such as approaching junctions, traffic signals, and taking suitable exits. A more sophisticated traffic layout would be that where other vehicles, in particular those in the subject vehicle's vicinity are themselves equipped with GA-INTACT systems carry out speed and lane changes in a highly dynamic manner, an remains as a future research challenge.

References

1. Arain, M.A., et al.: Action planning for the collision avoiding system using neural networks. Proc Intelligent Vehicles Sym. (1993) 119-124
2. Baluja, S., and Sukthankar, R.: Prototyping intelligent vehicle modules using evolutionary algorithms. Evolutionary Algorithms in Engineering Applications. Springer-Verlag (1998)
3. Campell, N.W., et al.: Autonomous road vehicle navigation. Engineering Applications of Artificial Intelligence. Vol.7(2), (1990) 177-190.
4. Ehlet, P.A.M., and Rothkrantz, L.J.M.: Microscopic traffic simulation with reactive driving agents. Proc. 4th IEEE Intelligent Transportation Systems Conf. USA (2001) 860-865
5. Foresti, G., et al.: A distributed approach to 3D scene recognition. IEEE Trans. Vehicular Technology. vol.43(2), (1994) 389-406
6. Goldberg, D.E.: Genetic algorithms in search, optimisation, and machine learning. Addison-Wesley (1989)
7. Liatsis, P., and Tawfik, H.M.: Two dimensional road shape optimisation using genetic algorithms. Mathematics and Computers in simulation. Vol.5, (1999) 19-31
8. Leutzbach, W.: Introduction to the theory of traffic flow. Splinger-verlag, Berlin (1988)
9. Lyons, G., and Hunt, J.: Traffic models - a role for neural networks. Proc. of neural networks and combinatorial traffic models. (1993) 71-79
10. Reece, D.A., and Shafer, S.A.: A computational model of driving for autonomous vehicles. Transportation research. vol.27A(1), (1993) 23-50
11. Sukthankar, R., Baluja, S., and Hancock, J.: Multiple adaptive agents for tactical driving. International Journal of Artificial Intelligence. Vol. 9(1), (1998)
12. Tawfik, H.M.: A graphical simulation environment for modelling of road and traffic scenarios. PhD Thesis. Control Systems Centre, UMIST, UK (2000)
13. Tribe, R.T., et al.: Intelligent driver support. 2nd World Congress on Intelligent Transport Systems. Yokohama, Japan (1995)

Agent-Based Mobile Robots Navigation Framework*

Wojciech Turek, Robert Marcjan, and Krzysztof Cetnarowicz

Institute of Computer Science
AGH University of Science and Technology
al. Mickiewicza 30, Krakow, Poland

wojciech.turek@agh.edu.pl, marcjan@agh.edu.pl, cetnar@agh.edu.pl

Abstract. The problem of mobile robot navigation has received a noticeable attention over last few years. Several different approaches were presented, each having major limitations. In this paper a new, agent-based solution the problem of mobile robots navigation is proposed. It is based on a novel representation of the environment, that divides it into a number of distinct regions, and assigns autonomous software Space Agents to supervise it. Space Agents create a graph, that represents a high-level structure of the entire environment. The graph is used as a virtual space, that robot controlling agents work in. The most important features of the approach are: path planning for multiple robots based on most recent data available in the system, automated collision avoidance, simple localization of a "lost robot" and unrestricted scalability.

1 Introduction

The most crucial features, that a mobile robot must possess, are ability of reaching desired position and determining current location in the environment. Moreover, the robot must be able to adopt to changes that take place in the environment, and to cooperate with other robots of the same or different type to avoid collisions or deadlocks. A number of approaches to this type of problems have been described, but each has some significant drawbacks or limitations.

Single robot navigation problems are known in literature as the "Simultaneous Localization and Map Building" (SLAM [1]) problem and the "Kidnapped Robot Problem". In the Kidnapped Robot Problem a robot "wakes up" in previously analysed environment; the task is to quickly estimate robot's position. In the SLAM problem a mobile robot is situated in an unknown environment, and is supposed to build a map of the environment basing only on own sensors readings. The map is simultaneously used for position estimation and path planning. Most popular approach to the SLAM problem[2][5][4] is based on determining position of a number of distinguishable landmarks (artificial or natural) and assumption of Gaussian distributions of observed landmarks locations errors - Kalman filters [3] are used. These solutions allow single robot to determine own location and

* This work was partially supported by the grant MEiN Nr 3 T11C 038 29.

to maneuver in a limited and usually invariable environment. Dynamic environments problem has also received a noticeable attention - Janglov [7] proposed a neural networks approach, in [8] Hahnel et al. presented a method of building a dynamic map with probabilistic model of obstacle presence certainty.

These solutions show, that abilities of a single robot are strongly restricted - its computational power, memory or effectors abilities may be insufficient, it cannot observe dynamic changes of whole large environment. Most of these problems can be theoretically overcome by a group of cooperating robots - memory and computations can be distributed, a few robots can perform one task. However, some new problems with navigation appear, mostly concerning movement coordination, group path planning and collision avoidance.

Several approaches were proposed. In [6] Arras et al. present a solution, that uses a graph for navigation in EXPO exhibition centre. The graph must be created manually, therefore the solution is not universal. A reactive collision avoidance method, where robots detect collision situation and modify their velocity to avoid it, is described in [9]. The method does not solve the problems of localization and multirobots path planning.

Lack of universal solutions to the problems of multirobot path planning and localization in dynamic environments makes it difficult to perform research into advanced applications of multirobots systems. Tasks like automated guarding, cleaning, transportation or manufacturing require reliable solutions of robots mobility problems.

Most research into navigation problems situates an autonomous robot in an unknown, hostile environment, and makes the robot responsible for gathering and processing data, that describe the terrain. The Framework proposed in this paper is based on a different assumption: fragments of environment, as well as robots, are given some consciousness and autonomy - they are considered to be software agents [10]. This approach makes it possible to apply global and local path planning algorithms that operate on a distributed and up-to-date knowledge about the environment. The solution supports robots with ability of reaching desired position at desired time, using optimal path and without causing collisions with other robots or obstacles.

2 Navigation System Architecture

The physical space, that robots operate in, is divided into relatively small rooms. Each room is supervised by a single Space Agent (SA), that is responsible for gathering and processing information about assigned terrain. Every Space Agent is supposed to know its direct SA neighbours - these connections create SA graph, which represents a high-level structure of the entire environment. The SA graph is also considered a virtual space, that mobile Robot Agents work in.

A Robot Agent (RA) is closely associated with a single physical robot. It controls robot's movements, receives sensor readings, and can perform type-specific operations. At any moment every RA is registered in one SA, depending on its robots location. When it needs to relocate, it informs the SA about desired

destination, and receives precise movement information. Simultaneously it sends to SA robot's sensors readings - this can be the only source of environment describing data. Due to autonomy and ability of communication, SAs are able to plan optimal, non-colliding paths for multiply robots, basing on most recent data available in the system.

This abstract architecture can be implemented in many different ways; two of them are described below.

2.1 Virtual Agents Scenario

More efficient and more demanding approach places software agents in a space, that is physically separated from the robots. The RAs and SAs are performing computations on computers in a network and control robots using radio connections. A communication infrastructure must cover entire environment, but processing can be faster and more complex when performed remotely. The architecture is presented in Figure 1.

There are many practical applications of this scenario, mostly concerning robots systems, that perform tasks like cleaning or guarding in the same place for a long time. An additional infrastructure is required, but also additional advantages appear. Devices like electric automated doors or elevators can be easily integrated with a corresponding SAs in order to increase robots functionalities.

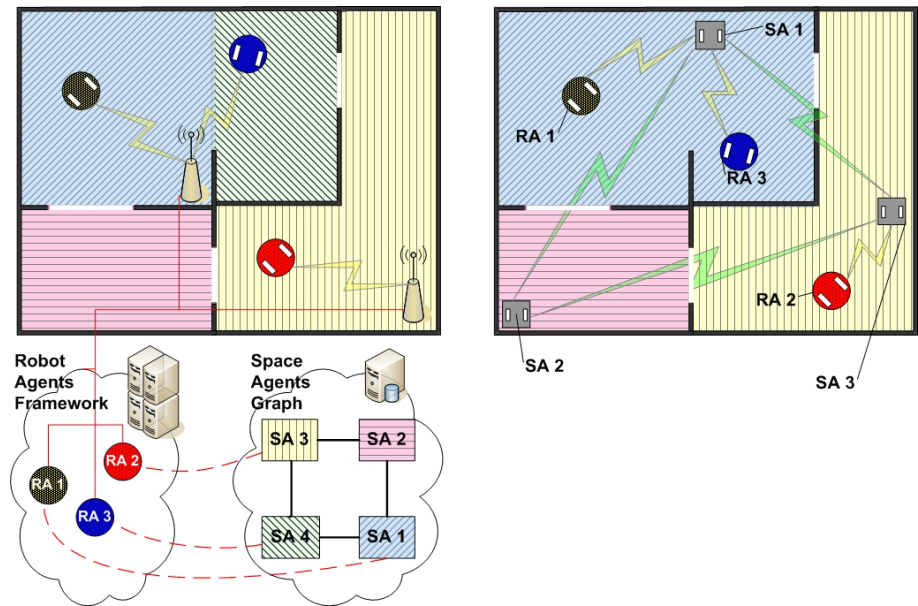


Fig. 1. Comparison of two implementations of the Navigation System: in an environment with communication infrastructure and external agent framework (on the left), and in an unknown environment with fully autonomous multirobot system (on the right)

Moreover, the situation does not restrict a size of the environment - neither memory nor computational power, that is required to navigate in a huge space, are limited by robots capabilities.

2.2 Hardware Agents Scenario

Described Navigation Framework can also be successfully implemented, when no external infrastructure is available in the environment. There are some additional requirements though. A set of robots is required to build a physical graph of Space Agents. The number and location of those robots is determined by communication ranges and environment properties - the SA graph must be consistent, every pair of neighbour SAs should establish a direct connection. Unlike the previous scenario, the robots must offer a significant computational performance, because all data storing and processing must be performed them. An example is shown in Figure 1.

3 Terrain Description Grid

Each Space Agent is responsible for a given part of the environment. It must gather data that describes the terrain from robots, that are passing through, and support these robots with information, that will help them to move fast and safely. The problem, that must be solved by an SA is very similar to a typical SLAM, except multiple sources of data are available. The solution presented below is a novel approach that can be used with any set of sensors, that can support sufficient amount of information. The assumption is made, that every robot has almost the same sensor readings, when situated in almost the same location.

A Space Agent applies a kind of graph, called a description grid, on the assigned space, as shown in Figure 2. Each node of the grid has unique location, and is connected with up to eight neighbours. Distance between nodes should be greater than a diameter of the smallest robot's bounding circle, so that two robots situated in adjacent nodes would not collide.

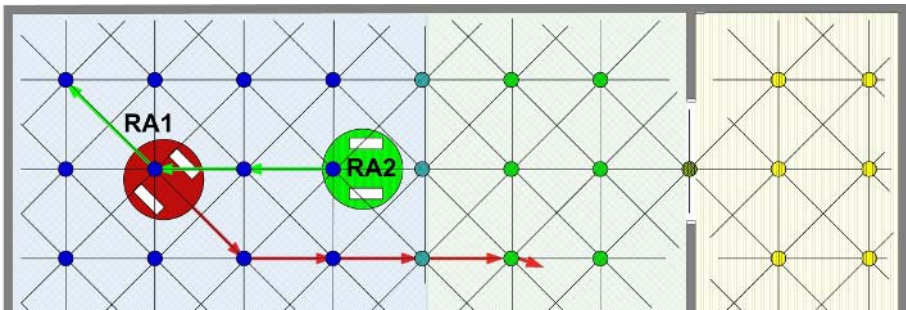


Fig. 2. Example of a terrain description grid, applied on a fragment of environment. Paths planned for two robots are not considered colliding.

Nodes are used for storing a vector of values returned by sensors of a robot, when situated in the node. Depending on types of sensors used, size of the vector will vary. General rules concerning selection of sensors are:

- two description nodes, that have very similar sensor readings vector should never be adjacent
- perfect (complete) set of sensors ensures that there are no two nodes in the environment that have the same sensor readings vector
- size of the vector should not be large, therefore sensor readings should be possibly uncorrelated
- a method for determining which nodes are physically accessible is required (e.g. directional range measurement).

The paradox of the SLAM problem is, that a map is built from robot's sensors readings gathered at a particular location, which are simultaneously used for determining this location. This issue does not cause problems as long as the environment (or at least landmarks set) remains static. If dynamic changes can occur, a robot may report a sensor reading vector, that is significantly different than the previously stored one. The SA must then decide, whether the environment has changed rapidly or the RA estimated wrong location of the robot.

Each RA knows an approximate position of controlled robot. The approximation accuracy is defined by a certainty factor $C_R \in [0, 1]$, which is reduced, as robot moves. When the robot reaches a node, its position is confirmed with SA, using stored sensor readings vectors, and robot's C_R is increased. Each sensor readings vector value that is stored at a node is also connected with a certainty factor $C_N \in [0, 1]$. If the readings vector is unknown: $C_N = 0$, just after a robot reports current vector: $C_N = 1$. The value is reduced as time passes, but once reported, never goes below 0.5.

If a conflict between stored and sensed readings occurs, the value with a higher certainty factor is considered real. When a robot "is right", node's data is updated, when all robots's readings are significantly different from the stored ones, so that its position cannot be confirmed, C_R is not increased. When C_R reaches 0, the robot is considered "lost" and kidnapped robot procedure is started.

There are two functions defined for each value in the sensor readings vector: inverted interpolation and propagation. The inverted interpolation function defines a method to calculate accurate robot's position basing on its sensor readings. If the description grid is dense enough, linear function can be used. The propagation function is used to update data in the nodes adjacent with the one the robot is located in - it is performed with reduced C_R value. The function depends on sensor type, and usually can be defined only for selected directions of adjacency.

4 Features of the Framework

The framework described in this paper solves several problems concerning mobile robots navigation:

Kidnapped robot problem. If a new RA is introduced to the system, it does not know its robot's location - the robot is considered "lost". Every SA in the system is then requested to find possible location, using robot's sensor readings vector. Provided, that the area, the robot is situated in, has already been covered by a SA's grid, location of the robot is estimated and robot's C_R is set to a value just above 0. If the location estimation is correct, the C_R factor will rise, otherwise the robot will soon become "lost" again.

SLAM problem. Starting with one RA with a robot at $(0, 0)$ and $C_R = 1$, and one SA, the map will be incrementally built, without any additional methods needed. The only requirement is, that SA should be able to grow its grid, and to divide if necessary. New robots can be added to the system, with a "lost" status, as soon as a part of the environment is covered by SA's grid.

Multiply robots navigation with collision avoidance. Navigation algorithm is divided into two stages:

- Global route planning - current SA searches the SA graph to find every possible route to a desired location. Each route is then split and analysed - the first, that meets given time requirements is selected.
- Local path finding - each SA in a particular route must plan robot's movements, so that it reaches the destination or moves to another SA. The SA receives initial location of the robot, time of reaching it, robot's size and maximum velocity, and desired destination coordinates. It is supposed to return guaranteed time of reaching the destination. If a SA has a complete information about assigned terrain, the local path finding can be considered a simple graph searching problem with constraints. Two types of constraints are possible: constant, when a node is physically inaccessible, and temporary, which occurs when a previously created path uses a particular node. Restricted nodes can be either omitted, or a robot may wait for node to become free.

After choosing the best route, a movement cycle begins:

- RA receives information about next location that should be reached by a robot
- robot moves, basing on previous location and some method of estimating covered distance (e.g. odometers readings, time and velocity calculations)
- sensor readings are sent to the SA in order to confirm current location and get next target coordinates
- SA verifies robots position and simultaneously updates information about the environment

Scalability. The environment description data is fully distributed, therefore scalability of the system is hardly restricted. If SA's memory or computational abilities turn out to be insufficient, it can always be divided, and each SA can work on a different computer in a network. The only restriction could be SAs

graph searching algorithm, but in most typical applications the number of SAs will not be greater than a hundred, which is few.

Up-to-date knowledge availability. One of the most important advantages of the approach is, that all navigation is based on complete and most recent knowledge, that is available in the system. Therefore planned paths are always most optimal.

5 Example of Implementation

First implementation of the Navigation Framework was tested with RoBOSS Simulation System [11]. A precise model of small soccer-playing robot is used; equipped with eight range finding sensors, an odometer and a compass. Nodes of the grid store readings from rangefinders; the data is used both for localization and for determining, which nodes are physically accessible. Manual methods for SA division are used; algorithms of path finding and lost robot localization are implemented.

The propagation function, started in a particular node, assigns an estimated values in nodes along rangefinders directions, with C_N reduced by 10% each node. The inverted interpolation function uses readings from nine nodes (3x3 square) to calculate eight lines, the robot should be situated on. Intersection points should be almost identical, indicating robot's accurate location; sum of distances between them is used for selecting best location of a "lost" robot.

Figure 3 shows a description grid and a SA graph generated for a simple maze. Three robots were able to reach randomly assigned locations without any problems, "lost" robot was also localized properly. Tests showed, that used set of sensors causes problems with accuracy, because even a very small error of compass readings causes large changes in values returned by rangefinders.

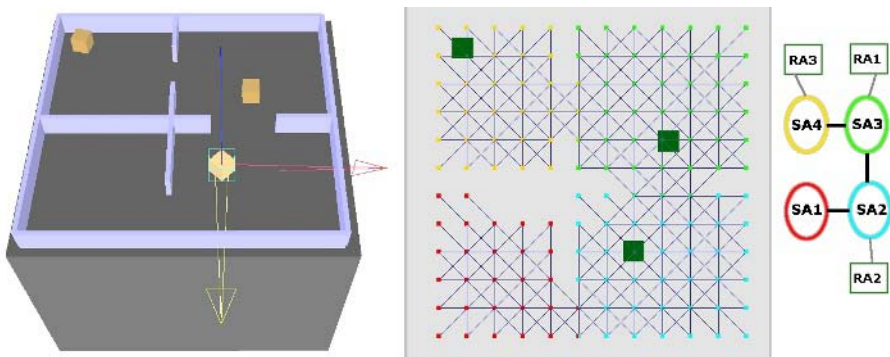


Fig. 3. Example of a simple environment with three FIRA MiroSot robots. A description grid, shown in the middle, is divided among four Space Agents.

6 Conclusions and Further Work

The Navigation Framework proposed in this paper solves several problems concerning multirobot navigation. The idea of Space Agents graph distributes the data that describes the environment, solving scalability problems. Path planning algorithm based on up-to-date information, guarantees optimal and collision-free solutions.

The example showed, that use of range finding sensors only, may be insufficient in some cases, therefore further research into types of sensors is required. Tests with real robots are also planned to verify the approach.

References

1. Clark S., Dissanayake G., Newman P., Durrant-Whyte H.F.: A Solution to the Simultaneous Localization and Map Building (SLAM) Problem IEEE Transaction. on Robotics and Automation, **vol. 17, no 3**, 229-241, 2001
2. Guivant J., Nebot E. Optimization of the simultaneous localization and map building algorithm for real time implementation. IEEE Transaction on Robotics and Automation, **vol. 17, no 3**, 242-256, 2001
3. Glielmo L., Setola R., Vasca F.: An interlaced extended Kalman Filter. IEEE Transaction on Automatic Control, **vol. 44, no 8**, 1999.
4. Montemerlo M., Thrun S., Koller D., Wegbreit B.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In Proc. of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002.
5. Se S., Lowe D., Little J.: Local and Global Localization for Mobile Robots using Visual Landmarks. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2001.
6. Arras K. O., Philippsen R., de Battista M., Schilt M., Siegwart R.: A navigation framework for multiple mobile robots and its application at the Expo.02 exhibition. In Proc. of the IEEE/RSJ IROS, Lausanne, Switzerland, 2002.
7. Janglov D. Neural Networks in Mobile Robot Motion.: International Journal of Advanced Robotic Systems, **vol. 1, no 1**, 2004.
8. Hahnel D., Triebel R., Burgard W., Thrun S.: Map building with mobile robots in dynamic environments. In Proc. of the IEEE International Conference on Robotics and Automation, 1557-1563, 2003
9. Madhava Krishna K., Hexmoor H.: Reactive Collision Avoidance of Multiple Moving Agents by Cooperation and Conflict Propagation. In Proc. of the IEEE International Conference on Robotics and Automation, NewOrleans, 2004
10. Cetnarowicz K.: M-Agent Architecture Based Method of Development of Multiagent Systems In Proc. of 8th Joint EPS-APS International Conference on Physics Computing, Krakow, Poland, 591-596, 1996
11. Turek W., Czyrnek D., Marcjan R., Cetnarowicz K.: RoBOSS - an universal tool for robots modelling and simulation. In Proc. of Computer Methods and Systems, Krakow, Poland, 347-354, 2005

The Autonomous Concurrent Strategy for Large Scale CAE Computation

P. Uhruski, W. Toporkiewicz, R. Schaefer, and M. Grochowski

Computer Science Department, AGH University of Science and Technology,
Kraków, Poland

uhruski@ii.uj.edu.pl, Wojciech.Toporkiewicz@sheraton.com,
schaefer@agh.edu.pl, grochows@ii.uj.edu.pl

Abstract. The paper presents the Agent-Oriented technology for running the parallel CAE computation. Fast and effective distributed diffusion scheduling is available that minimizes computation and communication time necessary for task governing and provides transparency in resource availability. Detailed evaluation of the diffusion rule parameters was obtained in the course of analysis of computational, memory and communicational complexity of CAE tasks.

1 Introduction

Computer Aided Engineering (CAE) tasks belong to the most tiring and resource consuming. The efficient way to solve such problems is the parallel processing in a distributed environment (see e.g. [14]). We suggest the Agent-Oriented approach for implementing and governing the CAE computation in the computer network. Such system delivers much more transparency and portability in comparison to the one designed by using the traditional tools (e.g. PVM, MPI). High efficiency of such a system is caused also by the special kind of diffusion scheduling [5] that allows to adopt dynamically the computing architecture to the current resource distribution and to minimize the communication overhead. Agents being parts of such a system called Smart Solids Agents (SSA) are specially designed to follow the Subdomain-By-Subdomain (SBS) method used as the parallelization strategy (see e.g. [15]).

2 CAE Concurrent Strategy Based on Non-overlapping Domain Decomposition

Let us consider a computational mechanics problem over a solid domain described by a differential equation (e.g. linear elasticity stationary problem). Application of *Finite Element Method* (FEM) makes possible to find its approximate solution by solving of a set of linear equations $Bx = f$. For FEM problems there is a parallel-processing oriented approach called Subdomain by Subdomain (SBS) [15] based on decomposition of Ω into a set of disjointed parts $\{\Omega_j\}$. SBS is usually combined with an iterative algorithm used for solution of resulting linear equation - *Conjugate Gradient* (CG) [16] in our case.

Domain decomposition (DD) on purpose of SBS is governed by two criteria: (i) meeting memory allocation constrains for a resulting subtask; (ii) minimization of interface between subdomains (see [12, 3]). Both of these criteria are evaluated in quantity of degrees of freedom (Dof) (see the next paragraph for Dof explanation) since on their number depend operating memory requirements and computation time of a resulting SBS task.

Generation of Computational Mesh (CM). CM in FEM consists of nodes grouped in elements [2]. Elements provide local support for base functions used to approximate problem's solution on the area of each of them. Dof are functionals connected with mesh nodes that define the local approximation of a solution over an element. Moreover each Dof corresponds with the single coordinate of the approximate solution x . Dof that are connected with CM nodes belonging to only one Ω_j are called internal Dof, while common to more than one - boundary Dof.

Generation of FEM matrices. As for matrix coefficients we utilize Galerkin's [15] formulation of FEM. Matrix of j^{th} subdomain is of the form:

$$\begin{pmatrix} B_{ii}^{(j)} & B_{ib}^{(j)} \\ B_{bi}^{(j)} & B_{bb}^{(j)} \end{pmatrix} \begin{pmatrix} x_i^{(j)} \\ x_b^{(j)} \end{pmatrix} = \begin{pmatrix} f_i^{(j)} \\ f_b^{(j)} \end{pmatrix}$$

$B_{ii}^{(j)}$ is a block of coefficients coming exclusively from internal Dof and $B_{bb}^{(j)}$ from boundary ones while $B_{ib}^{(j)}$, $B_{bi}^{(j)}$ are 'mixed' blocks.

Formulation of Schur Complement System (SCS). It is possible [15, 16] to transform a global problem to one comprising only boundary components of solution $x_b : Cx_b = f_b$ where $C = \sum_{j=1}^s (P_b^{(j)})^T C^{(j)} P_b^{(j)}$, $C^{(j)} = B_{bb}^{(j)} - B_{bi}^{(j)} (B_{ii}^{(j)})^{-1} B_{ib}^{(j)}$, $x_b = \sum_{j=1}^s (P_b^{(j)})^T x_b^{(j)}$, $f_b = \sum_{j=1}^s (P_b^{(j)})^T (f_{bb}^{(j)} - B_{bi}^{(j)} (B_{ii}^{(j)})^{-1} f_i^{(j)}) P_b^{(j)}$ are mappings between the global Dof numbering scheme used in Ω and local schemes used in $\{\Omega_j\}$.

Solution of SCS. Provided that SCS is symmetric and positive defined CG method may be applied to obtain its solution. CG is based on minimization of the quadratic form $q(x) = \frac{1}{2}x^T Cx - x^T f_b$ which has the unique minimizer x being also the solution of SCS. CG is an iterative method where starting from the initial point x_0 at each step approximate solution is improved in the direction $d_k : x_{k+1} = x_k - \alpha_k d_k, k = 0, 1, \dots$. Value for α_k can be determined by the explicit minimization of q along $x_k - \alpha_k d_k$ so that $q(x_k - \alpha_k d_k) = \min_{\alpha} (x_k - \alpha d_k)$. Iteration is terminated when the distance between x_k and x_{k+1} falls below required threshold (see e.g. [16]).

Finding "internal" variables. Having computed boundary variables x_b these referring to internal Dof can be evaluated using the formula $x_i^{(j)} = (B_{ii}^{(j)})^{-1} (f_i^{(j)} - B_{ib}^{(j)} x_b^{(j)})$.

Domain decomposition is implemented as a recursive bisection algorithm which consists in dividing problem's domain with planes into two parts in each step and greedy optimization of interface (see [3] for details). Domain decomposition is performed sequentially on one machine. Partitioning respects both criteria (i), (ii) and the density of Dof function ρ_{Dof} that provides the rough number of Dof falling into particular area Ω_j . This part delivers minimum computational complexity $O(\log s)$ with respect to the number of subdomains s . This task is performed sequentially and has negligible memory complexity.

We utilize tetrahedral Delaunay [2] CM generated in parallel over $\{\Omega_j\}$. Computational complexity of a task in this step is between $O(N^2)$ and $O(N^3)$ where N is the number of Dof in particular Ω_j . There is little communication between tasks - it is only necessary to interchange information on interfacing CM in order to assure its identity.

Computational complexity of FEM matrices generation may vary considerably depending on physical properties of a problem. There is no communication between tasks.

The key advantage of SBS consist in retaining the matrix C in a distributed form of its local components $C^{(j)}$. Finding each of $C^{(j)}$ may be accomplished by inverting local matrices $B_{ii}^{(j)}$ thus computational complexity of a task in this step is not greater then $O(N_{ij}^3)$ where N_{ij} is the number of internal Dof in particular Ω_j . There is no communication between tasks.

The main activity of a task during CG iteration consist in multiplication of a local component of SCS by subsequent approximation of the solution vector. Total computational complexity associated with the j^{th} subdomain is $O(\sqrt{n}N_{bj}^2)$ in this step where N_{bj} is the number of boundary Dof in Ω_j . \sqrt{n} represents a rough evaluation of PCG iteration number while n stands for the C dimension (see [1]). Total communication complexity is also $O(\sqrt{n}N_{bj}^2)$. This part of SBS process is strongly synchronized since after each multiplication step it is necessary to assembly entire vector x_k on the master node in order to verify coherence condition and compute direction vector d_k .

Finding internal variables can be completed in the time $O(N_{ij})$ and communication is $O(N_{bj})$ since only $x_b^{(j)}$ needs to be sent to the task associated with Ω_j . The maximum memory complexity of above steps is $O(N_{ij}^2 + N_{bj}^2)$.

Our implementation of SBS-CG process differs from the most standard approach in decomposing model of Ω on the base of ρ_{Dof} rather than partitioning ready CM. This provides for straightforward parallelization of later CM generation and since computational and memory complexity of this process is square with respect to the number of nodes this strategies provide considerable savings in terms of operating memory and CPU time.

3 The Course of the Agent-Oriented Approach

The agent paradigm may be applied to the CAE computation to unleash any possible relaxations in the course of computation. Each task associated with Ω_j part of a solid is being wrapped by an autonomous agent that tries to compute

as much independently as possible synchronizing with other tasks only when required. Agents are responsible for allocation of appropriate computational resources and running internal task work as long as no synchronization is required. Let us outline possible relaxation of consecutive steps of CAE technology:

- Mesh computation may be done autonomously with respect to the neighboring agents that need to exchange the sibling interface. That means this step requires only partial synchronization between neighboring solid parts and no global synchronization is imposed.
- Interior meshes are generated asynchronously.
- Linear equation matrix creation may be done autonomously while its assembling requires communication.

The proposed Agent-Oriented solution is based on the Octopus platform [4], which is composed of software servers statically allocated on computer nodes (hence named Virtual Computation Node - VCN) that perform information, migration and hibernation policies for mobile computing units - agents (see [7]). Every VCN maintains a basic set of operations supporting agents communication needs. It builds up the topology of neighboring VCNs that let agents examine load of sibling nodes and migrate if required using the diffusion based scheduling principle.

Such layered architecture makes the underlying network environment transparent to the agent based application. Octopus supports agent activities and provides required, unified information while hiding possibly heterogeneous environment including amount of machines, their load and network segments bandwidth. That allows agents to be executed in a heterogeneous environment. On the other hand, the computing application is composed of mobile agents that wrap computational tasks.

Octopus platform was implemented in JAVA [9] and on purpose of inter-agent communication utilizes CORBA [11] architecture.

4 Agent-Oriented CAE Computation in the Octopus Environment

The Smart Solid Agent (SSA) architecture [4, 8] was chosen to perform CAE type computational tasks. Its overall intention is to facilitate the design and implementation of computing application for heterogeneous and dynamic computer network.

Computing application composed of mobile SSA agents is responsible for execution of carrying task and for finishing its computation in the shortest possible time. Each Smart Solid Agent is represented by a pair $A = (T, S)$ where T is the carried task including all data required for computation and S is a shell responsible for the agent specific logic. The shell S maintains the computational task including communication capabilities and scheduling mechanism realized by partitioning of the agent and by migration among available computers in the network.

At the beginning the problems' domain Ω is decomposed into subdomains Ω_j to achieve optimal grain respecting the current computational environment. Then for every subdomain a computational task is created. In order to allow SSA to carry more than one task a special task called "task container" is utilized to wrap execution of many tasks. The agent partitioning is implemented by division of contained set of tasks and creation of two child agents with cloned shells and with new "task containers" including divided subsets of tasks.

5 Governing of the CAE Agents

The CAE multi-agent application starts from one agent containing all tasks coming from decomposition of the whole domain Ω of a CAE problem. The first agent as well as all its child agents search for needed resources by using the diffusion scheduling (see e.g. [5]). Roughly saying diffusion scheduling allows SSA to migrate to the least loaded VCN in its neighborhood or to be partitioned in case of insufficient resources found on the current VCN.

If an agent contains only one task it executes the task in its own thread. For a "task container" with more than one task their execution can be performed in parallel on UMA machines.

Diffusion rules implemented in agent logic S are based on the parameters E, M, C collected for all tasks working in the neighborhood of each particular VCN. Meaning of such parameters is as follows: E is the remaining computation time measured in units common for all computational tasks; M is required RAM and C characterizes communication needed for a task. Communication description C is a set of pairs $(T, data)$ where T is an identifier of a destination task and $data$ is amount of data exchanged between T and the task. The momentary values of E, M, C may be evaluated through analysis of memory, time, and communication complexity of utilized algorithms.

Each CAE task (see section 2) is responsible for generation of computational mesh over associated subdomains Ω_j , creation of FEM matrices and solving linear equations as a slave of a SBS-CG solver. Assuming featured complexities a task provides the following values required for performing diffusion scheduling, during each step of computation:

Mesh generation: $M \sim M_1 = \beta(N_{ij} + N_{bj})$ where β coefficient is dependent on the shape of a subdomain; $E \sim (N_{ij} + N_{bj} - N_{tj})^3$ where N_{tj} is the number of vertices already included in the final mesh. Mesh generator first creates all vertices and then takes one by one and generates final mesh (when the mesh is generated $N_{tj} = N_{ij} + N_{bj}$); C is insignificant and can be omitted. Communication is only required at the beginning of mesh generation to synchronize two-dimensional meshes on interfaces. The exact number of Dof is available when all mesh vertices have been generated. Before and during vertices generation step the number of Dof is evaluated as $(N_{ij} + N_{bj}) \simeq \int_{\Omega_j} \rho_{Dof}$.

Matrix generation: $M \sim M_2 = M_1 + (N_{jj}^2 + N_{bj}^2)$; $E \sim \alpha(N_{ij} + N_{bj})^2$ where α may be very large and may vary considerably depending on physical properties of a problem; There is no communication between tasks ($C = \phi$).

SCS formulation: $M \sim M_2$; $E \sim (N_{ij})^3$; $C = \phi$.

CG iterations: $M \sim M_2$; $E \sim \sqrt{n}N_{bj}^2(1 - (1 - q)^2tq^{t-2})$, $0 < q < 1$ where t is the CG iteration number; $C \sim \{(T_m, \sqrt{n}N_{bj}^2)\}$ and T_m stands for the master node of a SBS-CG solver.

Finding internal variables: $M \sim M_2$; $E \sim N_{ij}$; $C \sim \{(T_m, N_{bj})\}, j = 1, \dots, s$.

6 Numerical Test

Initially performed tests intend to show if the diffusion based strategy for the first two phases of CAE properly utilized available computational resources and provided satisfactory speedup. Our computing environment was composed of up to 10 PC machines with Intel Celeron processors and 512MB RAM. Every machine hosted one VCN application (an Octopus computation node). As the underlying operating system we selected the SLAX Linux operating system [10]. This Linux distribution was selected basing on the SLAX ability to boot up from a bootable CD. That way we were able to link an available IBM PC compliant machine into the computational platform without additional installation. Each booted machine completed a star-like Octopus virtual topology by connecting to one selected root machine. Hence we obtained a cluster-like environment of 10 machines all connected by a fast LAN network.

An example solid was selected (see figure 1) and pre-partitioned. That resulted in 14 tasks (12 similar in size and 2 slightly larger ones), each related to a

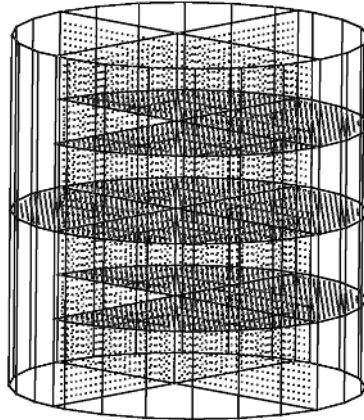


Fig. 1. Decomposed domain of the problem with visible interface meshes

Table 1. Speed up of mesh generation for different values of Dof function ρ_{Dof} representing mesh density

Mesh density	Number of vertices	Number of elements	Execution time [sec]		Speed up
			Serial	Parallel	
0,50	93420	535380	815	134	6,08
0,75	313523	1823385	5492	806	6,81
0,85	377562	2721561	10986	1595	6,89
1,10	975482	5823631	40613	5856	6,94

single piece of the solid. In the course of computations all machines were used by the agents thus 100% resources were allocated. The table 1 presents the speedup gained by the mesh generation agents versus serial computation time. Explanation of the achieved speedup comes from the number of tasks and the number of utilized machines. During the computation 6 machines were executing one task each (the larger tasks were executed by these), while other 4 machines were running two, similar in size, tasks each. We had 8, similar in size tasks executed in pairs by four machines. That means the maximum achieved speedup could be half of the tasks amount. Finally please note that the overall parallel time results from the the slowest tasks and since all tasks were similar in size, the overall speedup could be half of the tasks amount and that is 7.

7 Conclusions

The paper presents Agent-Oriented technology for running parallel CAE computations in a computer network. The approach provides fast and effective distributed diffusion scheduling that minimizes computation and communication time necessary for task governing. Moreover resources in a dynamic network environment are transparently available. (see [8, 5, 7] for more test results). Detailed evaluation of the diffusion rule parameters were obtained on the basis of computational, memory and communicational complexity analysis of CAE tasks. The numerical test presented in the section 7 shows behavior of agents performing two initial phases of CAE process - domain decomposition and mesh generation.

References

1. Barragy E., Carey G.F., Van de Geijn R.: Performance and Scalability of Finite Element Analysis for Distributed Parallel Computation. *Journal of Parallel and Distributed Computing* 21, (1994) pp. 202-212.
2. Georg P.L.: *Automatic Mesh Generation*. John Wiley & Sons, 1991.

3. Schaefer R., Toporkiewicz W., Grochowski M.: Rough partitioning of lumped structures, in *Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics*. Polish-Japanese Institute of Information Technology Press, Warsaw, October 2000, pp. 151-166.
4. Grochowski M., Schaefer R., Uhruski P.: An Agent-based Approach To a Hard Computing System - Smart Solid. *Proc. of the International Conference on Parallel Computing in Electrical Engineering (PARELEC 2002)*, 22-25 September 2002, Warsaw, Poland. IEEE Computer Society Press 2002, pp. 253-258.
5. Grochowski M., Schaefer R., Uhruski P.: Diffusion Based Scheduling in the Agent-Oriented Computing Systems. *Lecture Notes in Computer Science*, Vol. 3019, Springer 2004, pp. 97-104.
6. Momot J., Kosacki K., Grochowski M., Uhruski P., Schaefer R.; Multi-Agent System for Irregular Parallel Genetic Computations. *Lecture Notes in Computer Science*, Vol. 3038, Springer 2004, pp. 623-630.
7. Smoka M., Uhruski P., Schaefer R., Grochowski M.; The Dynamics of Computing Agent Systems. *Lecture Notes in Computer Science* Vol. 3516, Springer 2005, pp. 727-734.
8. Uhruski P., Grochowski M., Schaefer R.: Multi-agent Computing System in a Heterogeneous Network. *Proc. of the International Conference on Parallel Computing in Electrical Engineering (PARELEC 2002)*, 22-25 September 2002, Warsaw, Poland. IEEE Computer Society Press 2002, pp. 233-238.
9. Sun Microsystems, Java Technology, <http://java.sun.com/>
10. SLAX, SLAX Linux operating system, <http://slax.linux-live.org/>
11. CORBA, Object Management Group, <http://www.omg.org/>
12. Schaefer R., Flasiński M., Toporkiewicz W.: Optimal Stochastic Scaling of CAE Parallel Computations. *Lecture Notes in Computer Intelligence*, Vol. 1424, Springer 1998, pp.557-564.
13. Norton C. D., Cwik T. A.: Parallel Unstructured AMR and Gigabit Networking for Beowulf-Class Clusters. *Lecture Notes in Computer Science*, Vol. 2328, Springer-Verlag Heidelberg 2002, pp. 552-563.
14. Mann V., Parashar M.: Engineering an interoperable computational collaboratory on the Grid. *Concurrency and Computation. Practice and Experience.*, 14, pp. 1569-1593, 2002.
15. Papadrakakis M.: Domain decomposition techniques in Computational Structural Mechanics in M. Papadrakakis (Ed.), *Parallel Solution Methods in Computational Mechanics*, John Wiley and Sons (1996), pp. 87-140.
16. Golub G., Ortega J.M.: *Scientific Computing. An Introduction with Parallel Computing*, Academic Press Ltd., 1993.

Dynamic Resource Allocation Mechanism for Network Interconnection Management

Michał Karpowicz¹ and Krzysztof Malinowski^{1,2}

¹ Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

² NASK (Research and Academic Computer Network), Wawozowa 18, 02-796 Warsaw, Poland

M.Karpowicz@elka.pw.edu.pl, K.Malinowski@ia.pw.edu.pl

Abstract. We propose a dynamic resource allocation mechanism which can be used in multi-agent computer network interconnection management systems. Considering a setting of multiple consumers and elastic supply we argue that interaction between autonomous system resource managers is a game. We make the following contributions. First, we analyze the stability of the Nash equilibrium point of the resource allocation game. Second, we show that in comparison to the Cournot mechanism the mechanism we propose may lead to solutions which are characterized by a larger aggregate surplus.

1 Introduction

This paper is concerned with the dynamic resource allocation problem in a large scale, decentralized systems, namely bandwidth allocation problem between Internet service providers. It is well known, that if networks are to offer global services, service providers must cooperate and exchange Internet traffic. The obvious and intensively examined question that arises here is: how can network resources be allocated in an efficient way when network providers are most likely to act in their own self interest? We apply the game theoretic approach to this problem and propose a resource allocation mechanism which can be used in multi-agent computer network interconnection management systems.

Consider a group of interconnected network service providers and focus on dynamics of interactions between their *autonomous systems*. The basic observation we make suggests that since the number of interacting autonomous systems is small, local decisions concerning network resource allocations can have a significant influence on global network performance. Thus, decision makers, i.e. resource managers, may anticipate the effects of their actions on resource prices and view these prices as functions of the actions of all decision makers. This kind of interaction between autonomous system resource managers is a game.

We follow the approach applied by Johari and Tsitsiklis [3, 4, 5]. We analyze the resource allocation game through its *Nash equilibria* and design a simple market-clearing mechanism. The approach we apply is characterized by the three salient features. First, the monetary value of resource allocation is measured by

aggregate utility less aggregate cost. Second, the system sets the single price for resource unit to ensure demand equals supply. Third, agents anticipate the effect of their actions on market-clearing prices.

As we demonstrate in the following sections, the approach based on a simple market-clearing mechanisms may lead to solutions which are characterized by a very low computational and communicational complexity. From the network management point of view this features are crucial. Furthermore, results of Johari show that under reasonable assumptions the game we consider has a unique Nash equilibrium point. We use this results as a springboard for our mechanism design and propose a resource allocation mechanism which converges to this unique Nash equilibrium point.

This paper makes two contributions. First, we show global stability of the Nash equilibrium point of the game we consider. Second, we show that in comparison to the Cournot mechanism the mechanism we propose may lead to solutions that are characterized by a larger aggregate surplus. At this point we note that when agents are price anticipating Nash equilibria of the game we consider do not generally ensure full efficiency. Therefore, our propositions should be considered within the context of results of Johari and Tsitsiklis.

2 The Model

Following Kelly [1, 2] and Johari [3, 4, 5] suppose L agents compete for a single resource. Let a_i denote the rate allocated to agent i . First, we assume that:

Assumption 1. *For each $i \in L$, over the domain $a_i \geq 0$ the utility function $u_i(a_i)$ is concave, strictly increasing, and continuous, and over the domain $a_i > 0$, $u_i(a_i)$ is continuously differentiable. Furthermore, the right directional derivative at 0, denoted $u'_i(0^+)$, is finite.*

The total cost of using the amount $x = \sum_{i \in L} a_i$ of a resource is $c(x)$. We make the following assumption:

Assumption 2. *There exists a continuous, convex, strictly increasing price function $p(x)$ over $0 \leq x < C$ with $p(0) \geq 0$, such that for $0 \leq x < C$:*

$$c(x) = \int_0^x p(y)dy. \tag{1}$$

Thus total cost function $c(x)$ is strictly convex and strictly increasing over $0 \leq x < C$.

Suppose that the total payment agent i is willing to make for the amount a_i of resource is w_i . Let $w = [w_1, \dots, w_L]^T$.

Assumption 3. *For all $w \geq 0$, the aggregate allocation $x(w) = \sum_{i \in L} a_i(w)$ is the unique solution to $\sum_{i \in L} w_i = x(w)p(x(w))$. Furthermore, for each $i \in L$:*

$$a_i(w) = \begin{cases} w_i/p(x(w)), & w_i > 0 \\ 0, & w_i = 0 \end{cases} \tag{2}$$

Assumption 4. $p(x)$ is differentiable and exhibits nondecreasing elasticity $\varepsilon(x) = (\partial p(x)/\partial x) \cdot (x/p(x))$ for $0 \leq x < C$.

Now, consider a game in which, having $w_{-i} = [w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_L]^T$ fixed, player i selects $w_i \geq 0$ to maximize his payoff:

$$Q_i(a_i(w_i, w_{-i}), w_i) = u_i(a_i(w_i, w_{-i})) - w_i. \tag{3}$$

Agent i does not know w_{-i} , but he knows that the price $p(x(w))$ and $a_i(w)$ depend on it. Furthermore, he anticipates his own influence on the price. Denote the right directional derivative of $f(x)$ at x by $\partial^+ f(x)/\partial x$ and the left directional derivative by $\partial^- f(x)/\partial x$. The following results hold:

Proposition 1. (Johari) *If Assumptions 1-3 hold, then w is a Nash equilibrium of the game defined by $(Q_i(a_i(w), w_i))_{i \in L}$, if and only if $\sum_{i \in L} w_i > 0$, and for $a_i \equiv a_i(w) > 0$ and $x \equiv x(w)$ the following two conditions hold:*

$$u'_i(a_i) \left(1 - \frac{\frac{x}{p(x)} \frac{\partial^+ p(x)}{\partial x} a_i}{1 + \frac{x}{p(x)} \frac{\partial^+ p(x)}{\partial x} x} \right) \leq p(x); u'_i(a_i) \left(1 - \frac{\frac{x}{p(x)} \frac{\partial^- p(x)}{\partial x} a_i}{1 + \frac{x}{p(x)} \frac{\partial^- p(x)}{\partial x} x} \right) \geq p(x). \tag{4}$$

Conversely, if $a(w) \geq 0$ and $x > 0$ satisfy (4), and $x(w) = \sum_{i \in L} a_i(w)$, then the vector $w = [p(x(w)) \cdot a_1(w), \dots, p(x(w)) \cdot a_L(w)]^T$ is a Nash equilibrium.

Proposition 2. (Johari) *Suppose Assumptions 1-4 hold. Then there exists a unique Nash equilibrium \hat{w} for the game defined by $(Q_i(a_i(w), w_i))_{i \in L}$.*

Proposition 3. *Suppose Assumptions 1-4 hold and \hat{w} is Nash equilibrium of the game defined by $(Q_i(a_i(w), w_i))_{i \in L}$. Then the following conditions hold:*

$$(\forall i \in L) \quad p(x(\hat{w})) \leq u'_i(a_i(\hat{w})) \leq p(x(\hat{w}))(1 + \varepsilon(x(\hat{w}))). \tag{5}$$

Proof. Denote $\beta(x) = \varepsilon(x)/(1 + \varepsilon(x))$. Notice, that at equilibrium:

$$\begin{aligned} (\forall i \in L) \quad u'_i(a_i(\hat{w})) \left(1 - \frac{\beta(x(\hat{w}))}{x(\hat{w})} a_i(\hat{w}) \right) - p(x(\hat{w})) &= 0 \iff \\ u'_i(a_i(\hat{w})) - p(x(\hat{w})) &= u'_i(a_i(\hat{w})) \frac{\beta(x(\hat{w}))}{x(\hat{w})} a_i(\hat{w}) \geq 0 \end{aligned}$$

Now, suppose $u'_i(a_i(\hat{w})) > p(x(\hat{w}))(1 + \varepsilon(x(\hat{w})))$. Then we obtain:

$$\begin{aligned} (\forall i \in L) \quad p(x(\hat{w}))(1 + \varepsilon(x(\hat{w}))) \left(1 - \frac{\beta(x(\hat{w}))}{x(\hat{w})} a_i(x(\hat{w})) \right) &< p(x(\hat{w})) \implies \\ 1 + \varepsilon(x(\hat{w})) - \frac{\varepsilon(x(\hat{w}))}{x(\hat{w})} a_i(x(\hat{w})) &< 1 \implies \varepsilon(x(\hat{w})) \left(1 - \frac{a_i(x(\hat{w}))}{x(\hat{w})} \right) < 0. \end{aligned}$$

So we obtain that $(\forall i \in L) \quad a_i(\hat{w}) > x(\hat{w}) = \sum_{j \in L} a_j(\hat{w})$, which is a contradiction. □

3 The Mechanism

We use Proposition 1 and Proposition 2 to design a resource allocation mechanism which can be used in the resource negotiation process between agents.

Definition 1. We define operator $\mathcal{R}_{x_i}[f(x_1, \dots, x_N)]$ as follows:

$$\hat{x}_i \in \mathcal{R}_{x_i}[f(x_1, \dots, x_N)] \neq \emptyset \Leftrightarrow f(x_1, \dots, \hat{x}_i, \dots, x_N) = 0. \tag{6}$$

Definition 2. (Resource Allocation Mechanism)

Payment rule:

$$w_i^{k+1} = p^k \cdot \hat{a}_i^k \quad (\forall i \in L), \tag{7}$$

where

$$\hat{a}_i^k = \mathcal{R}_{a_i} \left[u'_i(a_i) \left(1 - \frac{\beta^k \cdot a_i}{x^k} \right) - p^k \right]. \tag{8}$$

Allocation rule:

$$a_i^{k+1} = \mathbf{1}(w_i^{k+1}) \cdot \frac{w_i^{k+1}}{p^{k+1}} \quad (\forall i \in L), \tag{9}$$

where

$$x^{k+1} = \mathcal{R}_x \left[\sum_{i \in L} w_i^{k+1} - xp(x) \right], \quad p^{k+1} = p(x^{k+1}). \tag{10}$$

The mechanism we introduce allocates resources according to Assumption 3, i.e. it sets the price $p(x(w))$ to ensure demand $\sum_{i \in L} w_i/p(x)$ equals supply x . Strategies w_i received by the mechanism denote *willingness* each agent has to pay for the rate a_i of resource at the unit price $p(x(w))$. These are equal to $p(x(w)) \cdot \hat{a}_i$, where \hat{a}_i maximizes payoffs $Q_i(a_i(w), w_i)$ at the price $p(x(w))$.

We now analyze convergence properties of mechanism (7) -(10) and examine the stability of the Nash equilibrium point \hat{w} .

Lemma 1. Let $\hat{p} = \sum_{i \in L} \hat{w}_i/x(\hat{w})$, where \hat{w} is Nash equilibrium of the game defined by $(Q_i(a_i(w), w_i))_{i \in L}$. If Assumptions 1-4 hold, then:

$$(p(x(w)) - \hat{p}) \left[\sum_{i \in L} \hat{a}_i(w) - a_i(w) \right] \leq 0. \tag{11}$$

Proof. Denote $p \equiv p(x(w))$. Suppose that $p \leq \hat{p}$. The optimal response to price p must lead to allocation for which the following condition holds: $\sum_{i \in L} \hat{a}_i(w) - a_i(w) \geq 0$. Conversely, if $p > \hat{p}$ then $\sum_{i \in L} \hat{a}_i(w) - a_i(w) < 0$. Now, condition $\sum_{i \in L} \hat{a}_i(w) - a_i(w) \geq 0$ implies that optimal price \hat{p} must be higher than or equal to $p = p(x(w))$, and if $\sum_{i \in L} \hat{a}_i(w) - a_i(w) < 0$ then $p > \hat{p}$. \square

Proposition 4. Let $\mathcal{F}(w(t)) = p(x(w(t)))\hat{a}(w(t)) - w(t)$, where:

$$(\forall i \in L) \quad \hat{a}(w(t)) = \mathcal{R}_{a_i} \left[u'_i(a_i) \left(1 - \frac{\beta(x(w(t))) \cdot a_i}{x(w(t))} \right) - p(x(w(t))) \right].$$

If Assumptions 1-4 hold, then Nash equilibrium point of the game defined by $(Q_i(a_i(w), w_i))_{i \in L}$ is an globally asymptotically stable equilibrium point of the system:

$$\dot{w}(t) = \mathcal{F}(w(t)). \tag{12}$$

Proof. First notice, that the only stationary point of system $\mathcal{F}(w(t))$ is $w = p(x(w))\hat{a}(w)$. Now, we show that $p(x(w(t)))$ converges to equilibrium price $\hat{p} = \sum_{i \in L} \hat{w}_i / x(\hat{w})$ when agents set their payments according to $\mathcal{F}(w(t))$. We have:

$$\begin{aligned} \dot{p}(x(w(t))) &= p'(x(w(t))) \sum_{i \in L} \frac{\partial x(w)}{\partial w_i} \dot{w}_i \leq p'(x(w(t))) \sum_{i \in L} \frac{p(x(w(t)))\hat{a}_i(w(t)) - w_i(t)}{p(x(w(t)))} = \\ &= p'(x(w(t))) \sum_{i \in L} (\hat{a}_i(w(t)) - a_i(w(t))). \end{aligned}$$

Consider dynamical system $\dot{p}(x(w(t)))$ and define the Lyapunov function $V(t) = [p(w(t)) - \hat{p}]^2$.

$$\begin{aligned} \dot{V}(t) &= 2[p(w(t)) - \hat{p}]\dot{p}(x(w(t))) \\ &\leq 2p'(x(w(t)))[p(x(w(t))) - \hat{p}] \sum_{i \in L} (\hat{a}_i(w(t)) - a_i(w(t))) \leq 0. \end{aligned}$$

So, for $w(0) = w_0$ price $p(x(w(t)))$ converges to \hat{p} . Under assumptions of the proposition this implies that $w(t)$ converges to \hat{w} . □

4 Comparison

Consider now the Cournot competition model and related resource allocation game with payoffs defined by:

$$\tilde{Q}_i(a, a_i) = u_i(a_i) - a_i p(\sum_{j \in L} a_j) \quad (i \in L).$$

We examine relations between allocations achieved at the Cournot equilibrium point \tilde{a} and at the Nash equilibrium point $a(\hat{w})$. We remind that $\tilde{a} = [\tilde{a}_1, \dots, \tilde{a}_L]$ is a Cournot equilibrium if:

$$(\forall i \in L) \quad \tilde{a}_i \in \arg \max_{a_i} \tilde{Q}_i(a, a_i).$$

Proposition 5. Consider payoff function $\tilde{Q}_i(a, a_i) = u_i(a_i) - a_i p(\sum_{i \in L} a_i)$. Let $\tilde{a}_i = \mathcal{R}_{a_i} [\partial \tilde{Q}_i / \partial a_i]$ and let \hat{w} be the Nash equilibrium of the game defined by payoffs $(Q_i(a_i(w), w_i))_{i \in L}$. If Assumptions 1-4 hold, then:

$$a_i(\hat{w}) \geq \tilde{a}_i \quad (\forall i \in L). \tag{13}$$

Proof. We show that at \tilde{w} such that $\tilde{a} = a(\tilde{w})$ we have:

$$\frac{\partial Q_i(a_i(\tilde{w}), \tilde{w}_i)}{\partial w_i} \geq 0.$$

We can write $\tilde{a}_i = \mathcal{R}_{a_i}[\partial\tilde{Q}_i/\partial a_i] = (u'_i(\tilde{a}_i) - p(\tilde{x}))/p'(\tilde{x})$. For $a(\hat{w}) = \tilde{a}$ from the equation $\partial Q_i(a_i(\hat{w}), \tilde{w}_i)/\partial w_i$, with $\tilde{x} = \sum_{i \in L} \tilde{a}_i$, we obtain:

$$\frac{u'_i(\tilde{a}_i)}{p(\tilde{x})} \left(1 - \frac{\beta(\tilde{x})}{\tilde{x}} \tilde{a}_i \right) - 1 = \frac{u'_i(\tilde{a}_i)}{p(\tilde{x})} \left(1 - \frac{u'_i(\tilde{a}_i) - p(\tilde{x})}{p(\tilde{x})(1 + \varepsilon(\tilde{x}))} \right) - 1.$$

Now, suppose that $a_i(\hat{w}) < \tilde{a}_i$. This implies that $u'_i(a_i(\hat{w})) > u'_i(\tilde{a}_i)$ and that:

$$\frac{\partial Q_i(a_i(\hat{w}), \tilde{w}_i)}{\partial w_i} = \frac{u'_i(\tilde{a}_i)}{p(\tilde{x})} \left(1 - \frac{u'_i(\tilde{a}_i) - p(\tilde{x})}{p(\tilde{x})(1 + \varepsilon(\tilde{x}))} \right) - 1 < 0.$$

We obtain:

$$1 - \frac{u'_i(\tilde{a}_i) - p(\tilde{x})}{p(\tilde{x})(1 + \varepsilon(\tilde{x}))} - \frac{p(\tilde{x})}{u'_i(\tilde{a}_i)} < 0 \Leftrightarrow \frac{u'_i(\tilde{a}_i) - p(\tilde{x})}{u'_i(\tilde{a}_i)} < \frac{u'_i(\tilde{a}_i) - p(\tilde{x})}{p(\tilde{x})(1 + \varepsilon(\tilde{x}))} \Leftrightarrow u'_i(a_i(\hat{w})) > u'_i(\tilde{a}_i) > p(\tilde{x})(1 + \varepsilon(\tilde{x})) > p(x(\hat{w}))(1 + \varepsilon(x(\hat{w}))).$$

This is a contradiction. According to Proposition 3 equilibrium allocation $a_i(\hat{w})$ can only occur when $u'_i(a_i(\hat{w})) \leq p(x(\hat{w}))(1 + \varepsilon(x(\hat{w})))$, so $\partial Q_i(a_i(\hat{w}), \tilde{w}_i)/\partial w_i \geq 0$. This proves that $a_i(\hat{w}) \geq \tilde{a}_i \ (\forall i \in L)$. \square

Proposition 6. Consider payoff function $\tilde{Q}_i(a, a_i) = u_i(a_i) - a_i p(\sum_{i \in L} a_i)$. Let $\tilde{a}_i = \mathcal{R}_{a_i}[\partial\tilde{Q}_i/\partial a_i]$ and let \hat{w} be the Nash equilibrium of the game defined by payoffs $(Q_i(a_i(w), w_i))_{i \in L}$. If Assumptions 1-4 hold, then:

$$\sum_{i \in L} u_i(a_i(\hat{w})) - c(\sum_{i \in L} a_i(\hat{w})) \geq \sum_{i \in L} u_i(\tilde{a}_i) - c(\sum_{i \in L} \tilde{a}_i). \tag{14}$$

Proof. First, notice that from Proposition 3 we have:

$$u'_i(\tilde{a}_i) \geq p(\sum_{i \in L} \tilde{a}_i) = p(\tilde{x}), \quad u'_i(a_i(\hat{w})) \geq p(\sum_{i \in L} a_i(\hat{w})) = p(x).$$

Furthermore, from the Proposition 3 and Proposition 5 we obtain: $u'_i(\tilde{a}_i) \geq p(x) = p(\sum_{i \in L} a_i(\hat{w}))$. Now, we have:

$$\begin{aligned} & \sum_{i \in L} \int_0^{a_i(\hat{w})} u'_i(a) da - \sum_{i \in L} \int_0^{\tilde{a}_i} u'_i(a) da = \sum_{i \in L} \int_{\tilde{a}_i}^{a_i(\hat{w})} u'_i(a) da \geq \\ & \geq p(x) \sum_{i \in L} \int_{\tilde{a}_i}^{a_i(\hat{w})} da = p(x)(x - \tilde{x}) \geq \int_{\tilde{x}}^x p(y) dy = c(x) - c(\tilde{x}), \end{aligned}$$

which completes the proof. \square

Proposition 5 shows that in comparison to the Cournot model the proposed mechanism (7)-(10) allocates more resources. Furthermore, Proposition 6 proves that it may lead to the higher aggregate surplus.

Figure 1 shows optimal response curves for two agents. Notice that since $a_i(a_j(w)) \geq a_i(a_j)$, at intersection point E^w both agents obtain more resources than at E^a . Figure 2 demonstrates convergence of the mechanism (7)-(10).

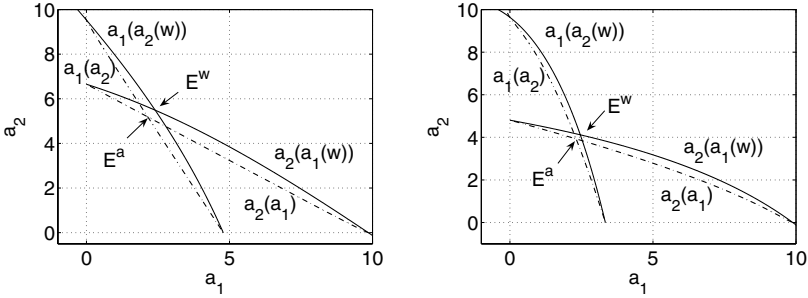


Fig. 1. Optimal response curves for $u_i(a_i) = \gamma_i \log(a_i + 1)$ (left), $u_i(a_i) = \gamma_i \arctan(a_i)$ (right) and $p(x) = (C - x)^{-1}$, $C = 10$

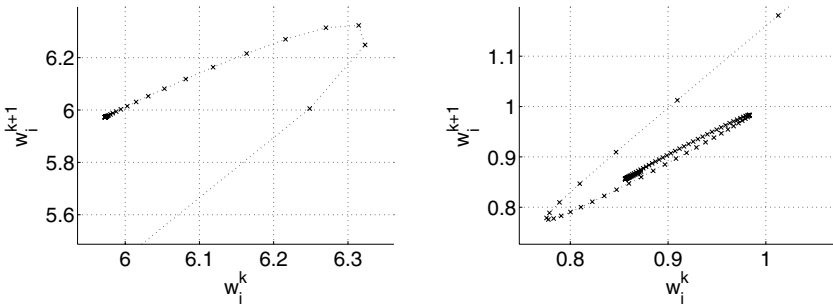


Fig. 2. Payment trajectories in phase space. Responses to p^k (left) and to $\sum_{k \in K} p^k / K$ (right) for $u_i(a_i) = \gamma_i \log(a_i + 1)$

5 Example

In this section we illustrate the results presented above. Suppose that L agents negotiate bandwidth allocations with local resource manager and that their utility functions are given by:

$$u_i(a_i(w)) = \gamma_i \log(a_i(w) + 1), \quad (\forall i \in L).$$

Suppose also that $p(x) = (C - x)^{-1}$, where C is effective bandwidth. Now, consider a provisioning problem. Suppose resource manager wants to sell a bandwidth of $C = 45$ Mbps to three groups of agents: $L_1 = \{1..10\}$, $L_{1.5} = \{11..20\}$ and $L_{25} = \{21, 22\}$. Group L_1 needs a peak rate of 1Mbps, group $L_{1.5}$ a peak rate of 1.5Mbps and group L_{25} a peak rate of 25Mbps. Let γ_i be random variables, independent and uniformly distributed on intervals $[3, 5]$, $[5, 7]$ and $[20, 21]$ for $i \in \{L_1, L_{1.5}, L_{25}\}$ respectively. We simulate the following scenario:

- $t < 0$: equilibrium state
- $t = 1$: agent $i = 21$ departs;
- $t = 2$: agent $i = 21$ joins, agents $\{11..18\}$ depart;

- $t = 3$: agents $\{1..10\}$ depart;
- $t = 4$: all agents join competition.

The scenario events take place when $\|a - \hat{a}\| < 0.00001$. We compare results obtained from mechanism (7)-(10) to results obtained from the Cournot mechanism. Notice that in the Cournot model agents submit allocations they want to buy at price $p(x(a))$. In case of mechanism (7)-(10) agents submit payments for allocations they need at price $p(x(w))$. Figures 3 present results of the simulation. Indeed, mechanism (7)-(10) allocates more resources than the Cournot mechanism and generates higher aggregate surplus.

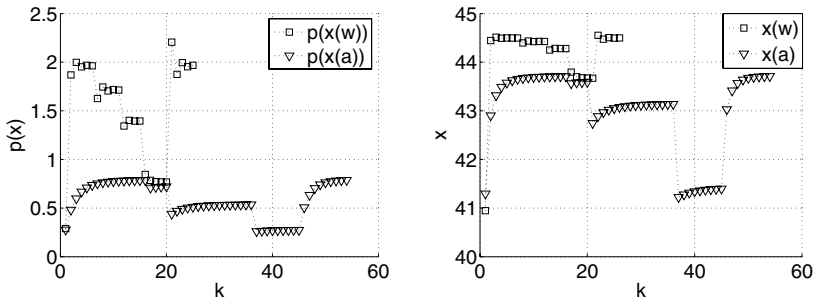


Fig. 3. Simulation results. Mechanism (7)-(10) allocates more resources ($x(w) = \sum_{i \in L} a_i(w) > \sum_{i \in L} a_i = x(a)$). This implies that $p(x(w)) > p(x(a))$. Notice that convergence rate of mechanism (7)-(10) is higher.

References

1. Kelly, F. P.: Charging and rate control for elastic traffic. *European Transactions on Telecommunications* **8** (1997) 33–37
2. Kelly, F. P., Maulloo, A. K., Tan, D. K.: Rate control for communication networks: shadow prices, proportional fairness, and stability. *Journal of the Operational Research Society* **49** (1998) 237–252
3. Johari, R., Tsitsiklis, J. N.: Efficiency loss in a network resource allocation game. *Mathematics of Operations Research* **29(3)** (2004) 407-435
4. Johari, R., Tsitsiklis, J. N.: Efficiency loss in Cournot games. Publication 2639, MIT Laboratory for Information and Decision Systems, 2005.
5. Johari, R.: Efficiency loss in market mechanisms for resource allocation. PhD thesis, Massachusetts Institute of Technology, 2004.

Computing MAS Dynamics Considering the Background Load

Maciej Smółka^{1,*} and Robert Schaefer²

¹ Institute of Computer Science, Jagiellonian University, Kraków, Poland
smolka@ii.uj.edu.pl

² Department of Computer Science,
Stanisław Staszyc University of Science and Technology, Kraków, Poland
schaefer@agh.edu.pl

Abstract. The paper extends the formal model of a computing Multi Agent System introduced in our previous papers to the case in which the background load coming from operating systems activities as well as other applications is included. Results concerning the existence of the optimal scheduling strategy as well as the characterization of such strategies have been obtained. The theorems partially verify scheduling heuristics (diffusion rules) designed and tested for large scale CAE computations.

1 Introduction

Multi-Agent Systems (MAS) dedicated to the distributed computing have been developed for the last several years (see. e.g. [1]). A formal model of such systems was introduced in our earlier papers [2, 3, 4]. It enables us to formulate a new definition of the optimal scheduling problem which is well suited to the diffusion-type distributed government of computing agents. We extend this model to the case in which the background load coming from operating systems activities as well as other applications is included. Results concerning the existence of the optimal scheduling strategy as well as the characterization of such strategies have been obtained. Such theorems partially verify scheduling heuristics (agent diffusion rules) designed and tested for large scale CAE computations [5, 6].

2 Properties of a Computing MAS

The suggested architecture of the system is composed of: *a computational environment* (MAS platform) and *a computing application* being a collection of mobile agents called *Smart Solid Agents* (SSA). The computational environment is a triple $(\mathbf{N}, B_H, perf)$, where:

$\mathbf{N} = \{P_1, \dots, P_N\}$, where P_i is a Virtual Computation Node (VCN). Each VCN can maintain more than one agent.

* This author has been supported by the State Committee for Scientific Research of the Republic of Poland under research grants 2 P03A 003 25 and 7 T07A 027 26.

B_H is the connection topology $B_H = \{\mathcal{N}_1, \dots, \mathcal{N}_N\}$, $\mathcal{N}_i \subset \mathbf{N}$ is an immediate neighborhood of P_i (including P_i as well).

$perf = \{perf_1, \dots, perf_N\}$, $perf_i : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a family of functions, which describes relative performance of all VCN's with respect to the total memory request M_{total}^i of all allocated agents. If M_{total}^i on P_i is small, $perf_i$ turns back the constant value, which depends only on the CPU architecture. If M_{total}^i is larger, $perf_i$ decreases due to the intensive swap utilization.

The MAS platform is responsible for maintaining the basic functionalities of the computing agents.

- It delivers the information about the local load concentration \mathbf{L}_j and Q_j (see (2) and (3) below),
- It performs the agent migration between the neighboring VCN's,
- It executes the agent partitioning and destruction,
- It supports the transparent communication among agents.

We shall denote an SSA by A_i where index i stands for an unambiguous agent identifier. Each A_i contains its computational task and all data necessary for its computations. Another component of an agent is a shell which is responsible for the agent logic. An SSA can denominate the pair (E_i, M_i) where E_i is the remaining computation time measured in units common for all agents of an application and M_i is the agent's RAM requirement in bytes. An agent may undertake autonomously the following actions:

- Continue executing its internal task,
- Migrate to a neighboring VCN if it finds better resources there (e.g. the new VCN is significantly less loaded),
- Decide to be partitioned, which results in creating two child agents $\{A_{i_j} = (T_{i_j}, S_{i_j})\}$, $j = 1, 2$. We assume that in the case of the agent partitioning the following conditions holds: $E_i > E_{i_j}$, $M_i > M_{i_j}$, $j = 1, 2$. The parent SSA dies after such a partition.

A computing application may be characterized by the triple $(\mathbf{A}_t, G_t, Sch_t)$, $t \in [0, +\infty)$ where:

\mathbf{A}_t is the set of application agents, $\mathbf{A}_t = \{A_{\xi_j}\}_{\xi_j \in I_t}$, I_t is the set of indices of agents active at the time t .

G_t is the tree representing agents partitioning at the time t . All agents constitute the set of nodes $\bigcup_{\xi \in \Theta} A_\xi$, $\Theta = \bigcup_{j=0}^t I_j$, while G_t edges show the partitioning history. All information on how to rebuild G_t is spread among all agents such that each of them knows only its neighbors in the tree.

$\{Sch_t\}_{t \in [0, +\infty)}$ is the family of functions such that $Sch_t : \mathbf{A}_t \rightarrow \mathbf{N}$ is the current schedule of application agents among the MAS platform servers. The function is represented by the sets w_j of agents' indices allocated on each $P_j \in \mathbf{N}$. Each of w_j is locally stored and managed by P_j .

Each server $P_j \in \mathbf{N}$ periodically asks all local agents (allocated on P_j) for their requirements and computes the local load concentration

$$L_j = \frac{E_{total}^j}{perf_j(M_{total}^j)} \text{ where } E_{total}^j = \sum_{i \in \omega_j} E_i \text{ and } M_{total}^j = \sum_{i \in \omega_j} M_i \quad (1)$$

Then P_j communicates with neighboring servers and establishes

$$\mathbf{L}_j = \{(L_\zeta, E_{total}^\zeta, M_{total}^\zeta, perf_\zeta)\} \text{ where } \zeta \text{ is such that } P_\zeta \in N_j \quad (2)$$

as well as the set of node indices Q_j such that

$$k \in Q_j \iff k \neq j, P_k \in N_j, L_j - L_k > 0 \quad (3)$$

The detailed description of the architecture and the agent governing strategies mentioned above as well as their current application may be found in our earlier papers (see e.g. [2, 5, 6]). The agent migration and partitioning rules described in these papers are related to the phenomenon of the molecular diffusion in crystals. The agent A_i migrates along the gradient of its binding energy which depends mainly on its remaining computation time E_i and the local load concentration [2].

3 The System State and the Background Load

In this section we recall our formal mathematic model for multi-agent computations. It was already presented in [2] and further developed in [3, 4], so we shall use the notations introduced therein. Denote by \mathbf{A} the set of all agents of an application. Recall the notion of the *vector weight of an agent* which is the mapping $w : \mathbb{N} \times \mathbf{A} \rightarrow \mathbb{R}_+^2$ whose components are E_i and M_i as above. Assume that the dependency of the total weight of child agents after partition upon their parent's weight before partition is well-known, componentwise and linear, i.e. we know the constants $c_1, c_2 \geq 0$ such that in the case of partition $A \rightarrow \{A_1, A_2\}$ we have

$$w_{t+1}^i(A_1) + w_{t+1}^i(A_2) = c_i w_t^i(A)$$

for $i = 1, 2$. Such an assumption seems realistic, in simple cases we can have $c_1 = c_2 = 1$ but in general the constants may be either greater or less than 1.

In order to avoid considering a single agent's dynamics we define a global quantity describing our computing system as a whole. Namely recall the notion of the *total weight of all agents allocated on a virtual node P at any time t* , i.e.

$$W_t(P) = \sum_{Sch_t(A)=P} w_t(A).$$

We put 0 if no agent is maintained by P .

In the sequel we shall assume that the number of virtual nodes $\sharp\mathbf{N} = N$ is *fixed*. Thus we can consider W_t as a nonnegative vector in \mathbb{R}^{2N} such that

$$W_t^j = W_t^1(P_j), \quad W_t^{N+j} = W_t^2(P_j)$$

for $j = 1, \dots, N$. We shall treat W_t as a *state of the computing MAS at a time t* . Considering its dynamics we assume first of all that in an idealized situation when migrations and partitions do not occur W_t evolves according to the following recursive equation

$$W_{t+1} = F(W_t, \xi_t) \quad (4)$$

where ξ_t is a given sequence of random variables (a stochastic process) representing the *background load*. (4) means that W_t is itself a discrete stochastic process. $W_t = 0$ means that our computations are finished so we do not want the system to leave this state. This leads to the assumption that for every t we have

$$F(0, \xi_t) = 0 \quad (5)$$

with probability 1.

As for ξ_t , we assume that it takes the finite number of values in the cube $[0, M]^N$. Its i -th coordinate $(\xi_t)_i$ may be interpreted as an additional load concentration on the i -th VCN, which may slow down computations performed by agents. We shall distinguish the following background load behaviors expressed by the conditions imposed on the sequence $(\xi_t)_{t=0,1,2,\dots}$.

- $\xi(1)$ The momentary background load changes rapidly many times during a single time period of the model. It may be generated by the basic activities of operating systems and network protocols. We assume that all ξ_t have the same distribution for each time period and they are mutually independent. This case has been considered in [4].
- $\xi(2)$ The variables ξ_t are also mutually independent but have different, well-known probability distributions. The distributions may change periodically after each T time steps of the model. Such a background load may be caused by small tasks which are executed in the period comparable with the time step of the model.
- $\xi(3)$ The sequence $(\xi_t)_{t=0,1,2,\dots}$ constitutes a stationary Markov chain. This is the case of the system additionally loaded by a set of similar tasks whose execution time is comparable to the several steps of the model. The large number of such tasks in a particular time step increases the probability of the high background load in the next step.
- $\xi(4)$ The variables ξ_t constitute a nonstationary, periodical Markov chain with period T . Such a situation may be related to the case of background tasks with the long execution time (much longer then the time step of the model) that appear according to a well-known trend. Such a situation is considered e.g. in [7].

4 The MAS Dynamics

Now we shall formulate the full equations of evolution of W_t . To this end we need to put into (4) terms representing migrations and partitions. A detailed

description of this process is presented in [4], here we show only the final step. Namely we have

$$\begin{cases} W_{t+1}^i = F^i(\tilde{W}_t, \xi_t) + c_1 u_{ii,t}^1(W_t) W_t^i + \sum_{j \neq i} u_{ji,t}^1(W_t) W_t^j \\ W_{t+1}^{N+i} = F^{N+i}(\tilde{W}_t, \xi_t) + c_2 u_{ii,t}^2(W_t) W_t^{N+i} + \sum_{j \neq i} u_{ji,t}^2(W_t) W_t^{N+j} \end{cases} \quad (6)$$

for $i = 1, \dots, N$, where

$$\tilde{W}_t^i = \left(1 - \sum_{k=1}^N u_{ik,t}^1(W_t)\right) W_t^i, \quad \tilde{W}_t^{N+i} = \left(1 - \sum_{k=1}^N u_{ik,t}^2(W_t)\right) W_t^{N+i}$$

with initial conditions

$$W_0 = \hat{W}. \quad (7)$$

$u_{jj,t}^m : \mathbb{R}_+^N \rightarrow [0, 1], m = 1, 2$ are the proportions of the weight components of splitting agents to the corresponding components of the total weight of all agents at node j at time t and $u_{jk,t}^m$ for $j \neq k$ are the respective proportions of the weight of agents migrating from j to k . Thus the first term on the right-hand side of (6) corresponds to the 'established' evolution of agents which neither migrate nor split, the second is the weight of partitioned agents and the third is the weight gathered as a result of the immigration. It follows that our W_t is a *controlled stochastic process* with a *control strategy*

$$\pi = (u_t^1, u_t^2)_{t \in \mathbb{N}} \quad (8)$$

such that $u_t^m : \mathbb{R}_+^{2N} \rightarrow U$. The control set U contains matrices $\alpha \in [0, 1]^{N \times N}$ that satisfy at least the following conditions

$$\alpha_{ij} \cdot \alpha_{ji} = 0 \text{ for } i \neq j, \quad \alpha_{i1} + \dots + \alpha_{iN} \leq 1 \text{ for } i = 1, \dots, N. \quad (9)$$

The first equation in (9) can be interpreted in the following way: *at a given time migrations between two nodes may happen in only one direction*. The second equality says that *the number of agents leaving a node must not exceed the number of agents present at the node just before the migration*.

In practice it may be appropriate to impose some stronger conditions on a control set (detailed considerations about that can be found in [4]). Denote by $G(W, u, \xi)$ the right hand side of (6) with $u = (u^1, u^2)$. Then we can rewrite our equation of evolution in the following way

$$W_{t+1} = G(W_t, u_t(W_t), \xi_t). \quad (10)$$

It is easy to see that thanks to (5) we have also that

$$G(0, u(0), \xi_t) = 0 \quad (11)$$

with probability 1 for any admissible u and any $t \in \mathbb{N}$.

In the most general case one could take the whole \mathbb{R}_+^{2N} as the state space S of W_t . But in reality both the available RAM and the maximal estimated time

of computations are bounded and quantized, so it is not unnatural to consider the state space *finite*.

The character of the state dynamics changes for different types of the background load. Detailed considerations on the case $\xi(1)$ are contained in [4]. In short, it can be shown that in this case W_t is a *stationary controlled Markov chain*. The case $\xi(2)$ is similar, but this time W_t is not stationary. In fact when ξ_t is T -periodic, W_t is a *T-periodic Markov chain* (for every u). In both cases (11) means that 0 is an *absorbing state* (cf. [8]) of W_t .

When ξ_t is itself a Markov chain, we can no longer say the same about W_t . We can only show that the latter is a *second-order Markov chain* (i.e. a process with two-step memory). Such a process can be transformed into a Markov chain by means of a simple classical transformation of the state variable (see e.g. [8]. Namely we should put

$$X_t = \begin{bmatrix} W_{t-1} \\ W_t \end{bmatrix}, \quad X_0 = \begin{bmatrix} \hat{W} \\ \hat{W} \end{bmatrix}. \tag{12}$$

It can be shown that the state equations for X_t have the form

$$X_{t+1} = \begin{bmatrix} X_t^2 \\ G(X_t^2, u_t(X_t^2), \xi_t) \end{bmatrix}. \tag{13}$$

In the case $\xi(3)$ such X_t is simply a stationary controlled Markov chain, and, again, in the case $\xi(4)$ it is T -periodic provided ξ_t is T -periodic. From the form of (13) and from (11) it follows that also in these cases 0 is an absorbing state.

Note that when W_t (or X_t) is T -periodic, it can be transformed into a stationary process by introducing a new state $Y_t = [W_{Tt}, \dots, W_{T(t+1)-1}]$. In the sequel we shall consider only stationary Markov processes, in other words the cases $\xi(1)$ and $\xi(3)$. But when we have a background process of type $\xi(2)$ or $\xi(4)$ which is periodic, we can apply the above transformation and all the following results shall remain true.

5 The Optimal Scheduling Problem

The general form of the cost functional for controlled Markov chains of our type is (cf. [8])

$$V(\pi; \hat{W}) = E \left[\sum_{t=0}^{\infty} k(W_t, u_t(W_t)) \right] \tag{14}$$

where π is a control strategy (8) and \hat{W} is the initial state of W_t . Since 0 is an absorbing state we shall always assume that $k(0, \cdot) = 0$, i.e. remaining at 0 has no cost (it guarantees that the overall cost can be finite). In the case $\xi(3)$ we need to rewrite (14) in the following manner.

$$\bar{V}(\pi; X_0) = E \left[\sum_{t=0}^{\infty} k(X_t^2, u_t(X_t^2)) \right] \tag{15}$$

Note that (15) inherits good properties (such as putting no cost on remaining at 0 or satisfying assumption A(3) of Prop. 1) from (14), so in the sequel we can safely substitute X_t for W_t and \bar{V} for V and all the results shall remain true.

We define the following set of admissible controls $\mathbf{U} = \{\pi : u_t^m(s) \in U_s, m = 1, 2\}$, where U_s is a *closed* subset of U (see (9)). Now we are ready to formulate the *optimal scheduling problem*. Namely given an initial configuration \hat{W} we search for such control strategy $\pi^* \in \mathbf{U}$ that

$$V(\pi^*; \hat{W}) = \min\{V(\pi; \hat{W}) : \pi \in \mathbf{U}, W_t \text{ is a solution of (10), (7)}\}. \quad (16)$$

Consider some cost functionals which seem appropriate for multi-agent computations. The first one is the expected total time of computations

$$V_T(\pi; \hat{W}) = E[\inf\{t \geq 0 : W_t = 0\} - 1]. \quad (17)$$

We can rewrite it in the form (14) if we put $k(s, \alpha) = 1$ for $s \neq 0$ and any α .

The second example takes into account the mean load balancing over time. It has the following form

$$V_L(\pi; \hat{W}) = E\left[\sum_{t=0}^{\infty} \sum_{i=1}^N (L_t^i - \bar{L}_t)^2\right] \quad (18)$$

where $L_t^i = \frac{W_t^i}{\text{perf}_i(W_t^{N+i})}$ is the load concentration and $\bar{L}_t = \frac{1}{N} \sum_{i=1}^N L_t^i$ is its mean over all computing nodes. This time the form of k is straightforward.

Both the above examples do not contain an explicit dependency on the control. Generalizing it a little allows us to penalize migrations. Namely take $\varphi : S \rightarrow \mathbb{R}_+$, $a \geq 0$, $\varrho_{ij}^m \geq 0$ and $\mu_{ij}^m : [0, 1] \rightarrow \mathbb{R}_+$ continuous nondecreasing and such that $\mu_{ij}^m(0) = 0$, and put

$$V_M(\pi; \hat{W}) = E\left[\sum_{t=0}^{\infty} \left(\varphi(W_t) + a \sum_{m=1}^2 \sum_{i \neq j} \varrho_{ij}^m \mu_{ij}^m(u_{ij,t}^m(W_t))\right)\right]. \quad (19)$$

6 Results

Now let us revisit some theoretical results that were proven for our model in [4] and extend them for considered types of the background load. First consider the existence of optimal solutions for problem (16). Let us recall that in [4] we used the notation $R(u)$ for the ‘probably not absorbing’ part of the transition matrix for our system while applying control u and by $R^n(u)$ the analogous part of n -step transition matrix obtained by applying the stationary control $u_t = u$ (for details on this notion see [4] or [8]). The following proposition is a straightforward consequence of [8, Theorem 4.2].

Proposition 1. *Assume that A(1) and (A(2) or A(3)) hold.*

- A(1) $k(s, \cdot)$ is continuous on $U_s \times U_s$
- A(2) $R^K(u)$ is a contraction for every u such that $u(s) \in U_s \times U_s$
- A(3) $R^n(u)$ is a contraction for some $n \geq 1$ and u as above but additionally $k(s, \alpha) \geq \varepsilon > 0$ for $s \neq 0, \alpha \in U_s \times U_s$.

Then there exists the unique optimal solution of (16).

This proposition allows us to prove the existence of optimal solutions for the cost functionals V_T , V_L and V_M . In case of V_T we need the assumption A(3), in case of V_L the assumption A(2) and in case of V_M either A(3) or A(2) depending on whether φ is separated from 0 or not (cf. [4]).

Now we shall recall some Bellman-type optimality conditions for (16). Let us denote the elements of the state space S by s_i , $i = 0, \dots, K$. The following proposition is another consequence of [8, Theorem 4.2] and its proof.

Proposition 2. *Make the same assumptions as in Proposition 1. Then the optimal solution of (16) is a stationary strategy $\pi^* = u^\infty = (u, u, \dots)$ and it is the unique solution of the equation*

$$V(\pi^*; s) = \min_{\alpha \in (U_s)^2} \left[\sum_{j=1}^K P(G(s_i, \alpha, \xi_0) = s_j) V(\pi^*; s_j) + k(s, \alpha) \right]. \quad (20)$$

The solution of (20) exists and it is the optimal solution of (16).

7 Conclusions

The presented MAS technology as well as the diffusion scheduling of agents give the advantageous environment for large-scale distributed computations (see [5]). The presented formal model based on the discrete stochastic field enables us to formulate a new definition of the optimal scheduling in such an environment. It has been shown (Propositions 1 and 2), that there exists an optimal diffusion scheduling strategy in the class of stationary rules that are intensively utilized during tests [5, 6]. This extends the earlier results [4] to the case in which the background load caused by operating systems and different kinds of other applications is taken into account.

References

1. Luque, E., Ripoll, A., Cortés, A., Margalef, T.: A distributed diffusion method for dynamic load balancing on parallel computers. In: Proceedings of EUROMICRO Workshop on Parallel and Distributed Processing, San Remo, Italy, IEEE Computer Society Press (1995) 43–50
2. Smolka, M., Grochowski, M., Uhruski, P., Schaefer, R.: The dynamics of computing agent systems. *Lecture Notes in Computer Science* **3516** (2005) 727–734
3. Grochowski, M., Smolka, M., Schaefer, R.: Architectural principles and scheduling strategies for computing agent systems. *Fundamenta Informaticae* (2006) accepted.
4. Smolka, M.: Optimal scheduling problem for computing agent systems. *Inteligencia Artificial* (2006) accepted.
5. Grochowski, M., Schaefer, R., Uhruski, P.: Diffusion based scheduling in the agent-oriented computing systems. *Lecture Notes in Computer Science* **3019** (2004) 97–104
6. Momot, J., Kosacki, K., Grochowski, M., Uhruski, P., Schaefer, R.: Multi-agent system for irregular parallel genetic computations. *Lecture Notes in Computer Science* **3038** (2004) 623–630
7. Lepiarz, M., Onderka, Z.: Agent system for load monitoring of the heterogeneous computer network. *Lecture Notes in Computer Science* **2328** (2002) 364–368
8. Kushner, H.: *Introduction to Stochastic Control*. Holt, Rinehart and Winston (1971)

Using Adaptive Agents for the Fault-Tolerant Mobile Computing System

Taeseon Park, Jaehwan Youn, and Dongryung Kim

Department of Computer Engineering, Sejong University,
Seoul 143-747, Korea
tspark@sejong.ac.kr, {rivian, fishangel}@sju.ac.kr

Abstract. This paper presents a fault-tolerance scheme based on mobile agents for the reliable mobile computing systems. The mobility of the agent is suitable to trace the mobile hosts in the system and the intelligence of the agent makes it efficient to support the fault-tolerance services. The proposed scheme especially focuses on the adaptiveness of the agent. The agents try to keep the failure-recovery cost and the failure-free operation cost within a certain level, regardless of the hand-off frequency of the mobile hosts. They also try to balance the two costs.

1 Introduction

Fault-tolerance is an important design issue to build a reliable mobile computing system and many fault-tolerance schemes have been proposed [2, 3, 5]. However, the fault-tolerance of these schemes is mainly supported by the mobile support stations of the system and hence these schemes have the following problems in common: First, the information required for the recovery of mobile hosts is managed by the mobile support station, which may cause the high processing delay of the support station. Second, for the efficiency of the recovery, the information managed by one support station is transferred to the new support station as the corresponding mobile host moves to the new area, which may cause the severe hand-off delay. Third, if the recovery information is not migrated to reduce the hand-off delay, the mobile host experiences a longer recovery delay in case of a failure.

To solve these problems, we have suggested a mobile agent based fault-tolerance scheme for the mobile computing system [4]. In this scheme, a stationary agent residing in each mobile support station site takes care of the recovery information of mobile hosts so that the support station can concentrate on its own tasks, such as the mobility handling and the communication of the hosts. Also, a set of mobile agents for each mobile host take care of the migration of the recovery information and hence the recovery information migration of a host can be performed asynchronously with its hand-off. As a result, the fault-tolerance service does not cause any unnecessary hand-off delay. Another notable point of this scheme is that mobile agents can make a migration decision suitable for the behavior of each mobile host.

However, in [4], the structure of the cooperative agents servicing the fault-tolerance and their interaction models are mainly discussed and not much attention is given to the migration strategy of the mobile agent carrying the recovery information. For the efficient fault-tolerance service, the migration decision of the agent is very important since the frequent migration may cause the severe failure-free operation cost and the lazy migration may cause the longer recovery delay of the host. Hence, in this paper, a new migration strategy of the mobile agent carrying the recovery information is proposed. The proposed scheme employs the adaptive agent which keeps evaluating the possible failure-recovery cost and the failure-free operation cost as the behavior of the mobile host, such as the hand-off frequency or the communication frequency, varies. Based on the evaluation, the agent schedules its migration to balance the two costs within a certain range.

2 Background

2.1 Mobile Computing System

A *mobile host (MH)* is an entity which executes the computation and communicates with another MH while traversing the system. In order to provide efficient and seamless services for MHs, the entire area covered by the mobile computing system [1] is divided into a number of service areas called the *cells* and one *mobile support station (MSS)* per cell is employed. Main tasks of the MSS are the location management and the seamless communication services for the MHs. For these services, MSSs are connected with each other through the high speed wired network; and when an MH leaves a cell and enters the next cell, two corresponding MSSs exchange information regarding the location and the communication status of the MH. This procedure is called the *hand-off*. The computation performed by the processes of an MH is assumed to follow a *piece-wise deterministic model*, in which a process can always produce the same sequence of states for the same sequence of message-receiving events.

2.2 Checkpointing and Message Logging

Checkpointing is an operation to save the intermediate states of the processes into the stable storage, so that the processes can restore the saved states and resume the computation from the restored states when the system fails [3]. Figure 1.(a) shows the effect of checkpointing. The horizontal arrow and the shaded boxes in the figure denote the progress of a process, P_i , and its periodic checkpointing, respectively. When P_i fails, it can resume the computation from the latest checkpoint, instead of restarting from the initial state, which is called the *rollback-recovery*.

Message logging is the operation to save the incoming messages of a process into the stable storage so that the process can re-execute the computation with the logged messages and reproduce the exactly same states which have occurred right before the failure [3]. Figure 1.(b) shows the effect of message logging. The

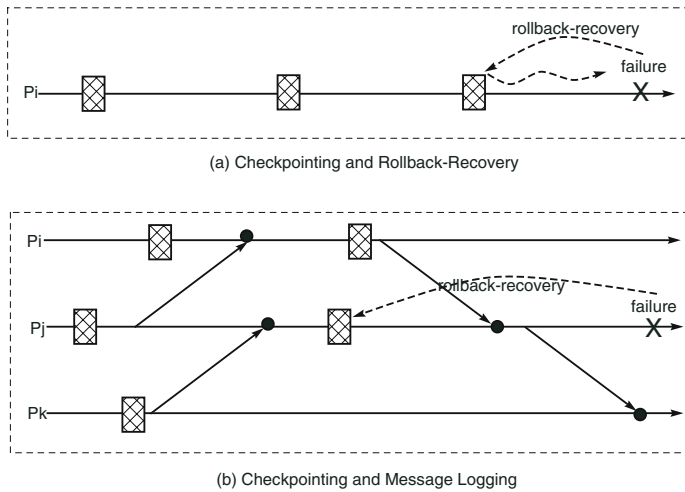


Fig. 1. Checkpointing, Message Logging and Rollback-Recovery

small black circle in Figure 1.(b) denotes the logging of each incoming message. With the message logging, P_j can re-process the logged messages during the recomputation, instead of receiving new messages, so that it can produce the same computational states which have been occurred right before the failure. As a result, the failure of P_j does not affect the other processes.

When the checkpointing and the message logging are considered for the mobile computing system, MSSs provide the stable storage since the storages of MHs are very limited. However, the MH moves around the system and hence checkpoints and message logs saved for an MH become dispersed as the MH moves over a number of cells. To efficiently manage the checkpoint and the message log, many approaches have been proposed [5].

3 Fault-Tolerance Service Based on Adaptive Agents

3.1 Basic Scheme

One common problem of existing schemes is to manage the checkpoint and the message logs as one pack. As shown in Figure 1, when a process fails, it first restores its latest checkpoint and then re-processes the logged messages. Hence, the checkpoint near the MH can help the early start of the recovery however the logged messages would be used gradually as the recomputation proceeds. Considering the heavy migration cost of the checkpoint and message log pack, it is wiser to separately manage these two types of recovery information. We have presented a basic design of mobile agent based recovery information management scheme in [4].

In the proposed design, three types of agents are used: One is the stationary agent, called a *Recovery Agent (RA)*. For each MSS site, one RA is used to

temporarily manage checkpoints and message logs produced by the MHs visiting the cell. The others are mobile agents, called the *checkpointing agent (CPA)* and the *log agent (LGA)*, each of which manages and makes the migration decision of the latest checkpoint and the message logs, respectively. The CPA retrieves the latest checkpoint from the RA and migrates to the local MSS site of the corresponding MH. The LGA retrieves the message logs from the RA or it can just record the log location to relieve the large burden of the log migration. Note that the migration of CPA and LGA is asynchronous with the migration of the corresponding MH.

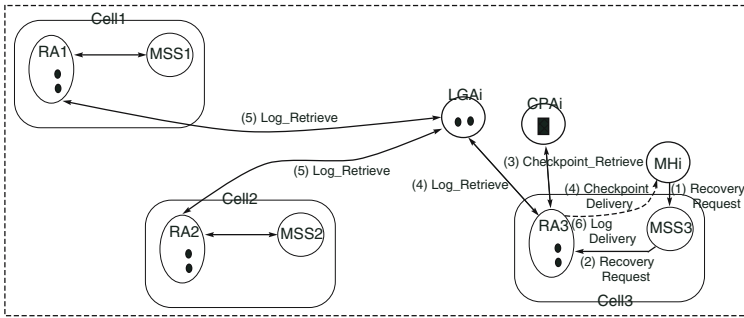


Fig. 2. Recovery based on Cooperative Agents

When an MH fails, the agents cooperate with each other to recover the failed MH. Figure 2 describes the recovery process taken by these agents. When MH_i in $Cell_3$ fails, the recovery agent, RA_3 , takes the prime responsibility of MH_i 's recovery. Hence, on the receipt of the *recovery* message from MH_i , RA_3 , contacts with CPA_i and LGA_i to retrieve the latest checkpoint and the logged messages for MH_i . On the request from RA_3 , CPA_i replies with the latest checkpoint and LGA_i sends the *log_collection* messages to the recovery agents recorded in its visiting list. After collecting the logged messages from other recovery agents, LGA_i , forwards the logs to RA_3 . For the early start of the recovery process, RA_3 starts the recovery of MH_i as soon as it receives the latest checkpoint from CPA_i , without waiting for the completion of the log collection. Then, RA_3 sorts the logged messages in the message sequence order and sends them to MH_i .

3.2 Adaptive Agent Based Fault Tolerance Service

One advantage of using mobile agents for the fault-tolerance service is that it is possible to design the efficient and proper agent migration strategy for each MH. Owing to a variety of hand-off rates, checkpointing intervals and communication rates of the MHs, the uniform migration strategy applied to all the MHs may result in the unpredictable and inefficient performance. Hence, to maintain the overhead caused by CPA and LGA migration to be in a predictable level, the careful design of the agent migration strategy suitable for each MH is desirable.

• **Distance-based scheme:** One important performance measure of this fault-tolerance service is the failure-recovery cost, which is the cost to retrieve the latest checkpoint and the message logs. The expected failure-recovery cost of MH_i , say C_{FR}^i , can be expressed as

$$C_{FR}^i = \alpha * D_{CPA}^i * S_{CPA}^i * C_N + \beta * \sum_{j=1}^{N_{log}} D_{log_j} * S_{log_j} * C_N, \quad (1)$$

where D_{CPA}^i is the expected distance between MH_i and its CPA at the time of a failure; S_{CPA}^i is the expected size of the CPA; C_N is the network cost for an object of the unit size to move in the unit distance; $D_{log_j}^i$ and $S_{log_j}^i$ are the expected migration distance and the expected size of each message log of MH_i ; and N_{log} is the expected number of logged messages to be migrated for each failure-recovery. The values of α and β are the weights. Since for the fast recovery, it is more important to retrieve the latest checkpoint than to collect the message logs, as it is discussed before, more weights should be put on the checkpoint retrieval.

Since our focus is not to analyze the performance but to select the major factors to affect the performance, we can simplify the failure-recovery cost as the function of D_{CPA}^i , such as

$$C_{FR}^i = k * D_{CPA}^i, \quad (2)$$

where $k = S_{CPA}^i * C_N$ and it is assumed that the values of S_{CPA}^i and C_N are constant throughout the execution of MH_i ; and all the weights are put on the checkpoint retrieval, such that $\alpha = 1$ and $\beta = 0$. Now, the system can maintain the constant or predictable failure-recovery cost for MH_i by adjusting the value of D_{CPA}^i . We call this scheme the *distance-based scheme* for the CPA migration. To implement the distance-based scheme, a proper value of D_{CPA} is first selected for the wanted level of C_{FR} . When MH_i migrates, it increments its migration counter; and it asks the *RA* for the migration of the *CPA* when the counter reaches the predetermined value.

• **Time-based scheme:** Another important performance measure of this service is the failure-free cost, which is the cost to migrate the CPA and the LGA (with or without its message logs) during the failure-free operation of the *MH*. Let C_{FF}^i be the expected failure-free cost of MH_i per unit time. Then, C_{FF}^i can be obtained as

$$C_{FF}^i = \frac{1}{T_{CPA}^i} * D_{CPA}^{i'} * S_{CPA}^i * C_N + \frac{1}{T_{LGA}^i} * D_{LGA}^{i'} * S_{LGA}^i * C_N, \quad (3)$$

where T_{CPA}^i and T_{LGA}^i are the time intervals between two consecutive migration events of the CPA and the LGA, respectively; and $D_{CPA}^{i'}$ and $D_{LGA}^{i'}$ are the expected distances between MH_i and its CPA or LGA for each migration, respectively. To simplify the equation, it is assumed that the values of S_{CPA}^i , S_{LGA}^i and C_N are constant throughout the execution of MH_i ; and we also assume the case that the LGA does not carry any message log. Under this assumption,

$S_{LGA}^i \ll S_{CPA}^i$ and hence the second term is ignored. Then, the failure-free cost can be rewritten as

$$C_{FF}^i = k' * \frac{1}{T_{CPA}^i} * f(T_{CPA}^i), \quad (4)$$

where $k' = S_{CPA}^i * C_N$ and $D_{CPA}^i = f(T_{CPA}^i)$. Note that the value of D_{CPA} is expressed as a function of T_{CPA} since the longer migration interval allows the more hand-offs of a MH.

Now, to maintain the constant and predictable failure-free cost regardless of the various behaviors of MHs, the *time-based scheme* for the CPA migration is suggested. In the time-based scheme, a proper time-out value, T_{CPA} is first selected for the wanted level of C_{FF} . The CPA queries the current location of MH_i and migrates to the corresponding MSS site, whenever the time-out timer expires. When the CPA moves, it calculates the moving distance and sets the next timer value as the value to $T_{old} * \frac{D_{old}}{D_{new}}$, where T_{old} is the previous time-out value, D_{new} and D_{old} are the current and the previous moving distances. As a result, the constant C_{FF} value can be managed.

• **Adaptive scheme:** However, when both of the failure-recovery cost and the failure-free cost are considered together, it is easily noticed that the two notions of time-based and distance-based schemes conflict with each other. When the short value of D_{CPA} is selected to reduce the C_{FR} , the value of $\frac{1}{T_{CPA}}$ should become larger, which causes the higher C_{FF} value; and vice versa. Hence, when the system wants to have low costs for both of the failure-recovery and the failure-free operation, the CPA should take one performance measure ignoring the other; or it should take a certain level of the costs in-between two wanted values.

For example, suppose that MH_i wants the value of D_{CPA}^i to be d and the value of T_{CPA}^i to be t . Let $C_{FR}^i[d]$ and $C_{FF}^i[d]$ be the failure-recovery cost and the failure-free cost when the distance-based scheme with the distance value of d is applied for the CPA migration. Similarly, let $C_{FR}^i[t]$ and $C_{FF}^i[t]$ be the costs when the time-based scheme with the time value of t is applied. Then, the adaptive CPA tries to keep the failure-recovery cost to be $\alpha * C_{FR}^i[d] + \beta * C_{FR}^i[t]$ and the failure-free cost to be $\beta * C_{FF}^i[d] + \alpha * C_{FF}^i[t]$, where $\alpha + \beta = 1$. Note that when $\alpha = 1$, the CPA chooses the constant failure-recovery cost; and when $\beta = 1$, the CPA chooses the constant failure-free cost. Otherwise, the CPA chooses the costs in-between the wanted ones.

To implement this adaptive migration strategy, the CPA basically follows the time-based method and the MH assigns the wanted values of T_{CPA} , D_{CPA} , α and β . The adaptive CPA first performs the time based migration with the given time-out value of T_{CPA} . For the first k migration, it observes the expected distance between the CPA and the MH, say D_{CPA}^E , under the given T_{CPA} and sets the target level of the distance as $\alpha * D_{CPA} + \beta * D_{CPA}^E$. Then, for each migration, the CPA examines the migration distance. If the migration distance exceeds the target distance, it decreases the value of T_{CPA} to reach the target distance. Since the adaptive CPA keeps adjusting the migration frequency

considering the expected recovery cost, the overall performance can be maintained within the expected level.

4 Performance Study

To measure the performance of the adaptive agent based fault-tolerance service, a simulation study has been performed. We have simulated a mobile computing system consisting of 100 mobile hosts and 1000 X 1000 rectangular cells [1]. One mobile support station is assumed for each cell. A mobile host stays in a cell for $\frac{1}{\lambda_n}$ time units and then migrates to one of its eight neighbor cells. The host communicates with another mobile host and the communication rate of each mobile host is $\lambda_m = 1/10$. The checkpointing rate and the failure rate of each mobile host are assumed to be $\lambda_c = 1/1000$ and $\lambda_f = 1/10000$. These rates are assumed to follow the exponential distribution.

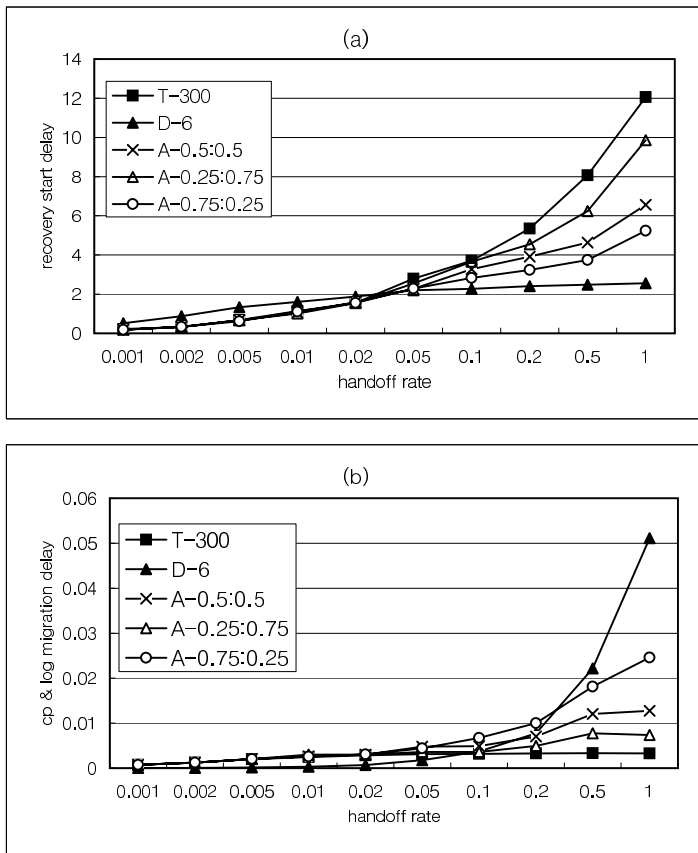


Fig. 3. Performance of the Adaptive Agents

Figure 3.(a) and (b) show the failure-recovery cost and the failure-free cost of the schemes, respectively. As it is expected, the distance based scheme with the distance value of 6, which is denoted by $D-6$, produces the constant failure-recovery cost while its failure-free cost is unpredictable. Also, the time based scheme with the time value of 300, which is denoted by $T-300$, guarantees the constant failure-free cost while its failure-recovery cost can be too high. The performance of adaptive schemes with different weights are denoted by $A-\alpha:\beta$ in the figure. The performance results show that the adaptive CPA effectively adjusts its performance for the various hand-off rates.

5 Conclusions

In this paper, we have presented the adaptive agent based fault-tolerance scheme for the reliable mobile computing system. The proposed scheme especially focuses on the adaptiveness of the agent which keeps the failure-recovery cost and the failure-free operation cost within a certain level considering the importance of the two performance measures. We also presented the simulation results supporting the effectiveness of the adaptive agents.

Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD)(KRF-2005-041-D00745) and also supported by grant No. B1220-0401-0037 from the University Fundamental Research Program of Ministry of Information & Communication in Republic of Korea.

References

1. Ayildiz, I.F., Ho, J.S.M.: On Location Management for Personal Communications Networks. *IEEE Communications Magazine* (1996) 138–145
2. Neves, N., Fuchs, W.K.: Adaptive Recovery for Mobile Environments. *Communications of the ACM*, Vol. 40, No. 1 (1997) 68–74
3. Park, T., Woo, N., Yeom, H.Y.: An Efficient Optimistic Message Logging Scheme for Recoverable Mobile Computing Systems. *IEEE Transactions on Mobile Computing*, Vol. 1, No. 4 (2002) 265–277
4. Park, T.: Mobile Agent based Fault-Tolerance Support for the Reliable Mobile Computing Systems. *Lecture Notes in Computer Science*, Vol. 3454 (2005) 173–187
5. Pradhan, D.K., Krishna, P., Vaiday, N.H.: Recoverable Mobile Environment : Design and Trade-Off Analysis. *Proc. of the 26th Int'l Symp. on Fault Tolerant Computing Systems* (1996) 16–25

A Multi-agent Approach to Resource Sharing Optimization in User Networks

J.C. Burguillo-Rial, E. Costa-Montenegro, and F.J. González-Castaño

University of Vigo
ETSET. Campus Univ. de Vigo
36310-Vigo, Spain
{jrial, kike, javier}@det.uvigo.es

Abstract. In this paper, we evaluate the feasibility of multiagent control of resources to be shared in user networks. A user network is totally controlled by the users, both at application and transport level. One of the possible applications in these networks is peer-to-peer (P2P) file exchange sharing the "external" access to the Internet (set of links between the user network and the Internet). If a node cannot serve its demand with its own external link, it requests help from another node via the high-bandwidth internal user network. We model user nodes as agents to simulate and to evaluate a new agent-based distributed control scheme. The simulation results in this paper confirm that it is possible to improve resource sharing in user networks using agents that take decisions autonomously, from local information, and check that file exchange services offered to neighbour nodes do not surpass appropriate credit limits.

1 Introduction

User networks are totally controlled by the users, both at application and transport level. This paradigm has become possible with the advent of broadband wireless networking technologies such as IEEE 802.11. For applications such as peer-to-peer file exchange [6], it may be useful to consider the "external" access to the Internet (set of links between the user network and the Internet) as a shared resource that can be optimized by node cooperation (i.e., if a node cannot serve its demand with its own external link, it requests help from another node via the high-bandwidth internal user network).

In this paper, we analyze the conditions that enable cooperation in user networks. This is not trivial in realistic scenarios. We could impose conditions leading to resource sharing via node cooperation that would never hold considering real user behaviour in peer-to-peer (P2P) services. We model resource sharing for P2P file exchanges in user networks as a game where node routers can cooperate or defect. Defection models "free-rider" users [14]: a typical problem in P2P networks.

The main goals of this paper are (1) to show that agent-based cooperative nodes may become a majority in user networks, improving resource sharing, and (2) those agent-based nodes can learn easily how to avoid free-riders by using adaptive credits.

The paper is organized as follows. Section 2 introduces user networks, peer-to-peer systems and some basic concepts of Game Theory. Section 3 presents the multiagent scenario and finally section 4 draws the conclusions.

2 User-Managed Networks, P2P and Game Theory

User-managed networks have become possible with the advent of wireless technologies such as IEEE 802.11 [4]. They represent one of the last stages in network control evolution [5]. This kind of infrastructures is currently being used to provide broadband access in Spanish rural areas, as an expansion of shared asymmetric DVB-S or DVB-T gateways. A typical basic node in a wireless user-managed network is composed by a router, an IEEE 802.11 access point (AP) and/or some IEEE 802.11 cards to set links with other basic nodes. Basic nodes may also be linked to a multi-user LAN (covering a building, for example).

A subset of the basic nodes will have cable or DSL access, providing "external" connection to the Internet. For the purposes of this paper, we will assume that all basic nodes are "externally connected". Additionally, we assume that user network capacity is larger than external access capacity (this holds for reasonable internal and external networking technologies, for example IEEE 802.11 and DSL respectively), so that the internal network always has spare capacity. In a user network, basic nodes can easily share contents, due to the large internal bandwidth. The bottleneck is the set of "external" connections to the Internet. By optimizing their usage, overall performance (and, as a consequence, user satisfaction) can be greatly improved.

By network stability we refer to the condition such that external demands (e.g., downloads or uploads at the edges of the user network for peer-to-peer file exchanges) can be satisfied with external capacity, on average. This certainly holds if:

1. The external demand of each basic node can be satisfied with its own external capacity, on average.
2. All basic nodes cooperate via the user network and their combined external demand can be satisfied with their combined external capacity, on average.

While cooperation is not strictly necessary to guarantee network stability, cooperation minimizes demand service time (nodes with temporarily idle external connections can help neighbours with demand peaks). However, there is no central authority, and probably selfish nodes will act to maximize their performance.

In general the P2P model improves the capability for resource sharing in any type of network. Applications based on such model include file-sharing systems (such as eMule [11], Kazaa [8] or BitTorrent [9]), discussion boards [12], overlay routing [16], etc. In these types of systems cooperation can be a useful strategy but it consumes node resources and performance. Since rational self-interested peers always try to improve their performance, they can refuse to supply service to others when they do not have clear incentives. Then, the "Tragedy of Commons" [13] may appear leading peers to generalized defection, i.e., free-riding [14].

Game theory [7] provides useful mathematical tools to understand the possible strategies that self-interested agents may follow when choosing a course of action. The context of cooperative games and cooperation evolution has been extensively studied in biological, social and ecological contexts [1], seeking general theoretical frameworks like the Prisoner's Dilemma (PD). In his seminal work, Axelrod has shown that cooperation can emerge in a society of individuals with selfish motivations [2]. For a review of related work in the last twenty years see [3]. Game Theory and the Generalized Prisoner's Dilemma have been applied to solve incentive

problems in P2P systems. Examples can be found in [15] and BitTorrent [9] itself that considers an alternative of the Tit-for-Tat strategy [2].

3 Multiagent Scenario

In this section we present a multiagent approach to support node decision. Now, we consider nodes ruled by agents that take their own decisions concerning the strategy they follow and the maximum credit limitation (since now *CreditLimit*) they allow to the neighbouring nodes for using their external connection. Therefore, *CreditLimit* defines the maximum amount of help (i.e., data) an agent would provide to a peer without receiving help in return.

The approach we follow in this paper is a composite spatial game where actions are effectively simultaneous but every agent may interact with several neighbours at a time. Every agent receives a data throughput payoff every turn. We consider throughput payoff as the amount of data send plus data received per unit of time. The better the strategy in its context, the better the payoff, i.e., the higher the throughput. Note that the payoff of a given agent depends on the choices made by the rest. After a pre-defined set of turns (24 hours = one day), each agent i chooses a strategy $s_i \in S$, where S is the set of all strategies. The agent will keep the strategy chosen to interoperate with its neighbours for the next day.

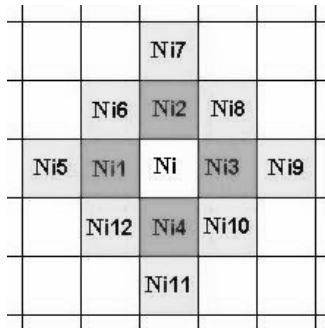


Fig. 1. Node N_i and two possible neighbourhoods: first with 4 nodes including $\{N_{i1}, N_{i2}, N_{i3}, N_{i4}\}$ and second with 12 nodes, from N_{i1} to N_{i12}

3.1 Spatial Distribution

For the spatial distribution of the nodes we consider a two-dimensional square lattice consisting of N nodes. Each node will follow one of the basic strategies (defection or cooperation). Figure 1 shows a cell node i with a strategy s_i and two possible neighbourhoods. In the model discussed in the next section, every cell (i.e. node) in the square lattice will interact with its neighbours to handle external traffic. Thus, there are both local and spatial interactions between neighbour cells. If we let every node in the system to interact with the remaining $N-1$ nodes, we have a panmictic population. But, in real user networks each node interacts with only a few (3 to 4)

neighbours. Thus, we consider that each cell i only interacts with the 4 cells in its immediate neighbourhood (N_{i1} to N_{i4} in figure 1).

Interaction is driven by demand service times of externally inbound or outbound data, i.e., when the external service queue of a node is larger than a particular threshold, the node contacts its neighbours requesting help to handle the files involved. In order to introduce a time scale, the time unit to generate new traffic demands and for neighbours' interaction is an hour. The total number of interactions per hour can be $4 \times N$ or less.

We decided that, during a 24-hour timeframe (a day), the strategy s_i of a node does not change. We also decided to consider a day as the time unit for strategy changes, since traffic patterns are similar along different days.

3.2 Strategy Types

We model three different strategies as follows:

- *Defection*: a defector never helps but ask their neighbours for help when it needs it. A defector models again a free-rider.
- *Cooperation*: a cooperator always helps its neighbours without taking care about any limitation. Cooperation models altruist users in P2P networks.
- *Agent-based*: a node following this strategy will adapt individually and dynamically its *CreditLimit* value to optimize the interaction with its neighbours.

We implement help transactions using the Contract Net Protocol: neighbours answer with a set of offers and/or refusals. The requesting agent selects the offering node that provided the best average throughput in the past.

3.3 Network Traffic

Concerning node demand distribution, we now define three node types A , B and C . These node types generate distributed demand service times (externally inbound or outbound data) during disjoint busy and quiet periods. The parameters are:

- A nodes: The busy period runs from 0 to 8 hours with uniformly distributed demand service times with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.
- B nodes: The busy period runs from 8 to 16 hours with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.
- C nodes: The busy period runs from 16 to 24 hours with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.

With these three node types we model three typical roles of Internet connection nodes: (A) for the late night users nodes (p.e., students' nodes), (B) for daily work users (p.e., commercial nodes); and (C) for leisure-time users (workers' home nodes). We also choose an average of 8 units during the busy period, which it is a little bit over the service time per hour (5 units) but not too much. The global service demanded per node in a day has an average of: $(8 \times 8 + 1 \times 16 = 80)$, which is less than $(24 \times 5 = 120)$. So we fulfil the two conditions described in section 2.

3.4 Simulation Scenario

We take a similar approach to the one presented in [10]. We consider that every agent stores a vector with the number of times NT_i that agent i has used every possible strategy. We define a strategy efficiency estimator to be:

$$EE_i(s,d+1) := \alpha \cdot f(i,d) + (1 - \alpha) \cdot EE_i(s,d) \tag{1}$$

Where $f(i,d) = \min_{h \in d}(th(h,i))$ represents, the minimum throughput value (th) obtained by agent i during any hour of that day. We consider the worst throughput as users try to improve their worst conditions. The α parameter is obtained from the formulae:

$$\alpha = w + (1-w) / NT_i(s) \tag{2}$$

Being $NT_i(s)$ the number of times that agent i uses the strategy s and w is a real-valued constant. We set ($w=0.3$) considering the results described in [10]. The term $(1-w)/NT_i(s)$ is a correcting factor, which has a major effect only when $NT_i(s)$ is low, when $NT_i(s)$ grows this term becomes negligible with respect to w .

To select the new strategy for the next day we need a probability distribution. Initially, we force every agent to test every possible strategy at least once. Then we do:

$$EN_i(s) = EE_i(s,d+1)^n \tag{3}$$

Where n is a positive real-valued parameter. To turn this into a probability we do for every strategy $s \in S$:

$$Prob_i(s) = EN_i(s) / ET_i \tag{4}$$

Being $ET_i = \sum_s EN_i(s)$ the normalization factor. Then clearly $EN_i(s)$ bias the selection towards strategies that have performed well in the past. The strength of the bias depends on n ; the larger the value of n , the stronger the bias. With high values of n (e.g., $n > 20$) the agent will always choose the strategy with the best record. But as explained in [15], this option does not allow the agent to explore other strategies when there are changes in the context. Therefore we set ($n = 10$).

3.5 Learning *CreditLimit*

Using agent-based strategy we want that every node learn what is the better *CreditLimit* it should apply considering its traffic conditions and the context where it is located, i.e., the surrounding neighbours and the interaction with them.

The agent has not too much time to explore the space of values of *CreditLimit*, since a delay in the convergence to the right value could cause throughput losses. In this framework, we consider that evolutive algorithms perform enough good and somehow simpler than other more sophisticated optimization techniques [17]. An evolutive algorithm considers a population that evolves on three phases: couple selection, crossover and sporadically mutation. This simple algorithm is defined as follows:

0. Every agent takes one random sample in every interval: [1, 10], [10, 20], [20, 30], [30, 40], [40, 50].
1. The agent chooses the best values (CL_1, CL_2) obtained and they become the selected couple. The other CL values are forgotten.

2. The newborn CL_3 is a linear combination (crossover) of its parents: $CL_3 = CL_1 + (1 - x) CL_2$ where $x = \text{rand}(0, 1)$.
3. Mutation: IF $(\text{rand}(0, 1) < 12 / \text{Hours})$ THEN $CL_3 = CL_3 + \text{rand}(-5, 5)$.
4. If CL_3 is better than CL_1 and CL_2 then the worst parent is replaced.
5. Return to step 2.

Note: In step 3, the first check is at 24 hours so the probability is lower than 1. We also limit the *CreditLimit* value range to [1, 50] after considering bigger intervals that were not used by the agents but introduce search delays.

3.6 Simulation Results

In this section we present the results obtained in the simulations performed with the Java UserNet simulator developed by the authors (access can be freely provided on demand). We considered a cell lattice of $(25 \times 25 = 625)$ cells in the user network. We also considered equally probable the initial selection of the 3 types of nodes and the 3 strategies. Every node neighbourhood has 4 nodes as described in figure 1.

Figure 2 shows the frequency evolution of the strategies along 50 days. The agent-based strategy becomes the more popular followed by the defector one. Cooperators still survive linked or surrounded by agent-based nodes as can be seen in the left side of figure 3, where darker cells are defectors, lighter ones cooperators and middle-dark agent-based nodes. In fact, agent-based nodes give somehow a skeleton to group cooperators around them with defectors surviving isolated and mainly exploiting cooperators. The value of *CreditLimit* learnt by the agents, had an average of 20.8 (in 10 runs) with a variance of 11.2 and most of the values fall in the range [5, 30].

Right side of figure 3 displays the throughputs achieved by the three strategies. At the beginning defectors have better results because they exploit cooperators and waste

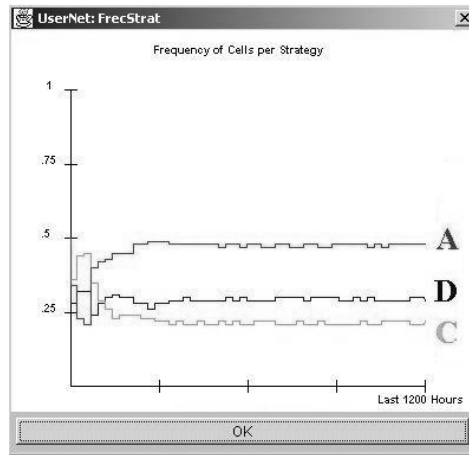


Fig. 2. Evolution of cooperators (C), defectors (D) and agent-based nodes (A) after 50 days

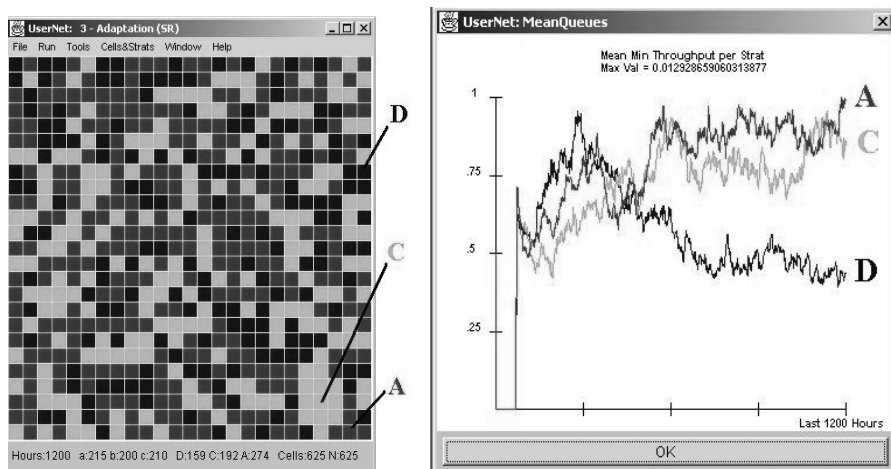


Fig. 3. Left: cell lattice state. Right: throughput comparison of cooperators, defectors and agent-based nodes. Time: 50 days; cooperators (C), defectors (D) and agent-based (A).

the *CreditLimit* provided by the agent-based nodes. Then, after 400 hours (around 17 days) cooperators and mainly agent-based nodes get better. At the end defectors perform half-better than the others.

Therefore, learning nodes chose the agent-based strategy as the most effective to avoid defectors (i.e., free-riding) and to improve their daily throughput.

4 Conclusions and Future Work

In this paper we present an abstraction of the problem of resource sharing in user networks. We have abstracted the main features of a real user network to obtain a simpler yet valid simulator. Then we show that under certain conditions (like learning and setting proper credit limits by every agent) cooperation becomes the most popular strategy, even in case of fully isolated and distributed node operation.

We point out that although the model finally looks simple, it has been far from trivial to be obtained. Thus, besides the model, the main contribution of this paper is to show that agent-based strategy nodes in user networks may become a majority, improving resource sharing, and avoiding free-riders by using adaptive credits.

Concerning future work, we think that these results can be extended to consider classical P2P file sharing over the Internet. In the short future we plan to design and implement a new protocol layer in user-networks to enable communication among all the user nodes. With such protocol, the agent coterie will not be restricted to its immediate neighbourhood providing global addressing within the user network (like the Internet). We also want to consider more realistic distributions for traffic and user network topologies. Finally, we consider interesting to extend the model to consider mobile user networks with dynamic connection establishment.

References

- [1] F. Schweitzer, J. Zimmermann, H. Muhlenbein: Coordination of decisions in a spatial agent model. *Physica A*, 303(1-2), 189-216, (2002)
- [2] R. Axelrod: *The evolution of Cooperation*. Basic Books, New York, (1984).
- [3] R. Hoffmann: Twenty years on: The evolution of cooperation revisited. *Journal of Artificial Societies and Social Simulation*, 3(2), (2000).
- [4] IEEE 802.11. [Online]. Available at the web site: <http://grouper.ieee.org/groups/802/11/>
- [5] J.P. Hubaux, T. Gross, J.Y.L. Boudec, M. Vetterli: Towards self-organized mobile ad-hoc networks: the terminodes project. *IEEE Commun. Mag.*, 1, 118-124, (2001)
- [6] Kazaa news. [Online]. 2004. Available at the web site: <http://www.kazaa.com/us/news/index.htm>
- [7] Ken Binmore: *Game Theory*. Mc Graw Hill, (1994).
- [8] Kazaa participation ratio. [Online]. 2005. Available at the web site: http://www.kazaa.com/us/help/glossary/participation_ratio.htm
- [9] The official BitTorrent page, <http://www.bittorrent.com>
- [10] A. Schaerf, Y. Shoham, M. Tennenholtz: Adaptive Load Balancing: A Study in Multi-Agent Learning. *Journal of Artificial Intelligence Research*, 2, 475-500, (1995)
- [11] Y. Kulbak, D. Bickson: The eMule Protocol Specification. [Online]. 2005. Available at the web site: http://leibniz.cs.huji.ac.il/tr/acc/2005/HUJI-CSE-LTR-2005-3_emule.pdf
- [12] Gu. B, and Jarvenpaa. S.: Are Contributions to P2P Technical Forums Private or Public Goods? - An Empirical Investigation, In *1st Workshop on Economics of Peer-to-Peer Systems*, (2003).
- [13] Hardin, G. *The Tragedy of the Commons*. *Science* 162, 1243-1248, (1968)
- [14] E. Adar and B. A. Huberman, *Free riding on Gnutella*. (2002).
- [15] Michal Feldman, Kevin Lai, Ion Stoica and John Chuang, *Robust Incentive Techniques for Peer-to-Peer Networks*. *ACM E-Commerce Conference (EC'04)*, (2004).
- [16] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and DS Wallach, *Security for Structured P2P Overlay Networks*. In *Proceedings of Multimedia Computing and Networking*, (2002).
- [17] U.M. García-Palomares, F.J. González-Castaño, J.C. Burguillo-Rial. *A Combined Global & Local Search (CGLS) Approach to Global Optimization*. *Journal of Global Optimization (JOGO)*. (article in Press, beginning of 2006).

Heterogeneous Behavior Evaluations in Ethically–Social Approach to Security in Multi-agent System*

Gabriel Rojek¹, Renata Cięciwa², and Krzysztof Cetnarowicz³

¹ Department of Computer Science in Industry
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
rojek@agh.edu.pl

² Department of Computer Networks
Nowy Sącz School of Business — National-Louis University
ul. Zielona 27, 33-300 Nowy Sącz, Poland
rcieciwa@wsb-nlu.edu.pl

³ Institute of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cetnar@agh.edu.pl

Abstract. Ethically–social approach to security problem uses the idea of agents functioning evaluation in a multi–agent system in the analogous way to the evaluation of a person’s behavior in small human societies. This approach involves distributed evaluations made by autonomous agents and processing of the results of this evaluations in order to create a collective decision of a society of agents. Research presented in this paper focuses on the part of domain of ethically–social behavior evaluation that specifies how an agent evaluates the behavior of another agent. The idea of heterogeneous behavior evaluations is formulated and tested in this domain.

1 Introduction

The main idea of security solutions design, that can self–adjust to new and unknown threat, is the use of a model of security mechanisms that function in everyday life and prevent from misusing an individual by others in his environment. Mechanisms inspiring our work are based on the evaluation of observable results of agents’ behavior. In a society every individual observes and evaluates the behavior of all other observable persons. Results of autonomously made behavior evaluations form one decision of the whole society e.g. decision to exclude somebody from a group. Security mechanisms in a society have the decentralized and distributed character — all individuals make their own autonomous evaluations which result in forming one decision of the society.

* This work was partially supported by the Ministry of Education and Science grant no. KBN 3 T08B 042 29.

Having a goal to design security mechanisms in a multi-agent system that are similar to that functioning in human societies, two problems have to be solved. The first problem is the design of evaluations mechanisms which each agent will be equipped with. These mechanisms should enable an agent to evaluate behavior of another agent functioning in a society. Each agent undertakes actions which are perceived in a computer system as objects. That objects registered in a certain period of time create a sequence of actions, that could be processed in order to qualify whether it is a *good* or a *bad* acting agent in the particular system, where evaluation takes place. The second problem is managing, collecting and processing the results of autonomous behavior evaluations in order to state if a particular agent, that is differently evaluated by different agents, is generally *good* or *bad* agent (called also an *intruder*).

The main topic of this paper is the tuning of behavior evaluation mechanisms of an agent to obtain the most infallible decision mechanisms of the whole security system. As it was observed in our earlier works (e.g. [1, 2]) it could incidentally happen that an agent with undesirable behavior was not identified by a society of evaluating agents as *bad* and, as a consequence, this agent was not excluded from that society. On the other hand, it could also happen that a *good* agent could be mistakenly treated as an *intruder*. In order to present stated problems, security mechanisms are presented: in Sect. 2, behavior evaluation mechanisms that are build in every agent in the system and in Sect. 3 mechanisms of management, collecting and processing the results of behavior evaluations. In Sect. 4 the influence of some coefficient's value of behavior evaluation mechanisms is analyzed on the decision of the whole group of agents. In Sect. 5 the idea of heterogeneous behavior evaluation is presented, tested and discussed. The main conclusions of this paper are stated in Sect. 6.

2 The Division Profile

All security mechanisms in which every agent is equipped are named the *division profile*. An agent that posses the division profile can evaluate behavior of an other agent that is visible in an environment of a multi-agent system. Every agent in a secured multi-agent system possesses his division profile, what means that in the system there are made many autonomous behavior evaluation.

Algorithms of the division profile are inspired by immunological mechanisms of T cells generation which enable to detect some anomalies. Usage of immunological mechanisms is restricted and the aim of our work is not to present an artificial immune system. In a case of behavior evaluation, immunological intruders detection mechanisms have to operate on observed actions made by evaluated agent. This approach is opposite to the one proposed in e.g. [3, 4] in which immunological mechanisms operate on the structure of resources. An other difference between artificial immunology and ethically-social approach is the autonomy of a process (an agent) in a secured system. In artificial immunology approach one detection system is considered for a particular computer system (or sub-system). In ethically-social approach every agent uses his own instantiation of

detection mechanisms autonomously, what induces the necessity of application of some additional algorithms in order to agree collective decision of all agents.

According to immunological mechanisms of T cells generation the division profile has 3 stages of functioning: creation of collection of *good* (*self*) sequences of actions, generation of detectors and behavior evaluation stage.

2.1 Collection of Good Sequences of Actions

In order to use mechanisms of intruder detection with an immunological approach, the collection of *good* sequences of objects representing undertaken actions has to be specified. The collection of *good* sequences of actions corresponds to the collection of *self* elements in an artificial immunology system. In ethically-social approach to security problem it is possible to assume that an agent evaluates his own behavior as *good*. Taking into consideration this assumption, the collection W of *good* sequences of actions of an agent consists of sequences of actions undertaken by this agent. The length of a sequence is fixed to l . Presuming h last actions undertaken by every agent are stored, his own collection W will contain $h - l + 1$ elements.

In order to generate the collection W an agent should collect information representing actions undertaken by him in the past. But, on the other hand, an agent in order to evaluate behavior of another agent has to collect information representing actions undertaken by the evaluated agent. So an agent should have information about all actions made in the system. This information is stored in the table of actions. In our work every agent stores last h actions of every visible agent in his table of actions.

2.2 Generation of Detectors

Detectors are used to evaluate behavior in the analogous way to the functioning of T-lymphocytes in the immune system. In presented work detectors of an agent are generated only once in the whole time of existence of this agent. Generation of detectors of an agent happens, when this agent "knows" his last h actions for the first time, so when this agent has undertaken h actions (collection W has to be completed before detectors' generation).

The algorithm of detectors' generation uses the negative selection — from set R_0 of generated sequences of length l those matching with any sequence from collection W are rejected. Set R_0 contains every possible sequence (but it is also possible to use a set of sequences generated at random). The length of a detector is equal to the length of a sequence from collection W . Sequence matching means that elements of those sequences are the same. Sequences from set R_0 which will pass such a negative selection create a set of detectors R .

2.3 Behavior Evaluation Stage

Once detectors of an agent have been generated, this agent can evaluate behavior of another agent. The result of behavior evaluation process of an evaluating agent a is a coefficient attributed to an evaluated agent k . This coefficient marked as m_a^k is a number of counted matches between:

- detectors of the agent a which evaluates behavior,
- sequences of actions undertaken by the agent k (this sequences of actions are taken from table of actions of the agent a).

Marking the length of a detector as l and the number of stored actions as h , the coefficient m_a^k is a number from a range $\langle 0, h - l + 1 \rangle$. The maximum of counted matches is equal $h - l + 1$, because every fragment of sequence of actions, which has length equal to the length of a detector, can match only one detector.

3 Mechanisms of Distributed Evaluations Agreement

An algorithm of agents evaluations managing, collecting and processing is used to agree one common decision of all agents which assess behavior evaluations. The difficulty of this agreement is caused by the fact that an agent can be differently evaluated by different agents in the system. This problem is presented in [5]. Only information essential to discuss the main topic of this article are presented in this section. In presented research the power function of behavior evaluation is used.

Each action undertaken by an agent may cause change of the results of behavior evaluations that are done by other agents in the system. This approach lets us formulate *the algorithm of evaluation management* as follows: If agent k undertakes an action, a request of the evaluation of agent k is sent to all agents (except agent k) by the environment.

Agent a in case of receiving a request of evaluation of an agent number k sends back only the coefficient o_a^k in the range $0 \leq o_a^k \leq 1$. The coefficient o_a^k is given by function:

$$o_a^k = \left(\frac{m_a^k}{h - l + 1} \right)^4 \tag{1}$$

where $h - l + 1$ is the maximum of counted matches of agent a . The power function of evaluation behavior increases the weight of high coefficient m_a^k (the exponent was set empirically).

In order to decide if agent k is in general *good* or *bad*, the environment uses *the algorithm of evaluation collecting and processing*, which consists of following actions:

1. All results of behavior evaluations are stored and, are sent by agents in response to the request of evaluation of the agent k .
2. Gained coefficients are summed and then this sum is divided by $j - 1$ (j is the number of agents):

$$o_*^k = \frac{o_1^k + o_2^k + \dots + o_{k-1}^k + o_{k+1}^k + \dots + o_{j-1}^k + o_j^k}{j - 1} \tag{2}$$

If o_*^k is greater than $\frac{1}{2}$ agent k is eliminated.

4 Tuning of the Division Profile

In this section our aim is to present the influence of the main parameters of behavior evaluation mechanisms on the process of evaluation. Parameters of functioning of division profile are: l — the length of detectors, h — the number of actions stored in table of actions. The quality of ethically–social behavior evaluations has two dimensions: the speed of intruders removing and the rate of self–destruction phenomenon (the number of *good* agents mistakenly evaluated as intruders).

An experimental multi–agent system was implemented in order to discuss the influence of main parameters of division profile on the quality of evaluation process. Two types of resources are in the simulated environment: resources of type A and resources of type B. Resources are used by agents, but refilling of all resources is only possible when each type of resources reaches the established low level. This specific refilling of resources reflects the situation in real computer system with some operations which must be executed in couples, for example: opening and closing a file, connection request and disconnection request. There are a lot of attack techniques that are limited to only one from a couples (or trios...) of obligatory operations (e.g. SYN flood attack [6]). The simulated system has three types of agents:

- 50/50 agents — agents which take one unit of randomly selected (A–50%, B–50%) resource in every full life cycle;
- 90/10 agents — agents which take one unit of randomly selected (A–90%, B–10%) resource in every full life cycle; type 90/10 agents can be treated as intruders, because the increased probability of undertaking only actions of one type can block the system;
- 100/0 agents — agents which take one unit of A resource in every full life cycle; 100/0 agents are also called intruders.

Actions of 90/10 agents are to certain degree similar to actions of 50/50 agents but they are also undesirable in the secured system. The division between 50/50 agents and 90/10 agents is hindered by random character of agents decision which resource select. It could happen that a *good* 50/50 agent takes (much) more units of one resources than units of another resource in certain time period.

4.1 The Self–destruction Phenomenon

A multi–agent system with initial 90 agents was run in order to research the self–destruction phenomenon. All of the agents are 50/50 *good* agents. Measure of the self–destruction phenomenon is the percentage of agents remained in this system after 1000 constant time periods Δt . Because results could differ in every run of the simulation (random character of agent’s decision making), the simulation was run 10 times. Average of obtained results are presented in Table 1.

The results for length of detector $l \geq 6$ for number of stored actions $h \leq 20$ indicate that near all agents are deleted because of the self–destruction phenomenon. This dependence is caused by the fact that for so long detectors the

Table 1. Percentage of agents removed after 1000 constant time periods Δt in consequence of self-destruction phenomenon with the coincidence of parameters h — number of stored actions and l — length of detectors

-	$h = 16$	$h = 17$	$h = 18$	$h = 19$	$h = 20$...	$h = 27$	$h = 28$	$h = 29$
$l = 4$	0.22%	0.22%	0.22%	0.00%	0.00%	...	0.00%	0.00%	0.00%
$l = 5$	17.78%	7.11%	2.78%	1.33%	0.44%	...	0.00%	0.00%	0.00%
$l = 6$	97.78%	97.78%	7.78%	97.78%	97.78%	...	0.89%	0.22%	0.22%
$l = 7$	97.78%	97.78%	97.78%	97.78%	97.78%	...	97.78%	97.78%	97.78%
$l = 8$	97.78%	97.78%	97.78%	97.78%	97.78%	...	97.78%	97.78%	97.78%

information stored in the table of actions was not precise enough to generate usable set of detectors. In process of generation of detectors a negative selection is used as it is presented in Sect. 2.2. In the case of not enough information about actions done in the past (to small parameter h), not enough elements are excluded from the set R_0 and detectors become elements of the set R_0 which should be excluded in process of negative selection. Agent a possessing such big set of detectors evaluates all kinds of behavior with great coefficients o_a which means evaluating all kind of behavior as *bad* and as a result self-destruction of agents. However, in the case of $l = 5$ should be stored $h \geq 18$ actions in order to restrict the self-destruction phenomenon, in the case of $l = 6$ the table of actions has to be extended to $h \geq 27$ in order to ensure information to detectors' generation process. Conclusions presented in this paragraph are confirmed by experimental measured number of detectors with the coincidence of parameters h and l (results not presented in this paper).

4.2 The Rate of Intruders Detection

In order to measure the speed of intruder detection a multi-agent system was simulated. At the beginning of each simulation 90 *good* 50/50 agents were in the system. In 50-th constant time periods Δt one intruder was put into the system. The rate of intruders detection is the time needed to remove this agent. As obtained results of experiments show the removing of 90/10 agents is more problematic than the removing of 100/0 agents. 100/0 agents are removed faster and more faultlessly, so only results of the removing of 90/10 agents are presented as more sensitive to parameters tuning. Obtained results are presented in Table 2 (the simulation was run 10 times for each presented case).

Taking into consideration results presented in Table 1 and in Table 2, it could be stated that for length of detectors $l = 5$ the optimal number of actions stored in the table of actions is $h = 18$, however greater number of stored actions involves restriction of the rate of the self-destruction phenomenon. This dependency can be explained by the fact that more information about behavior (more undertaken actions) involves more accurate behavior evaluation. On the other hand more information required to evaluate a behavior (e.g. $h = 19$) needs more time to obtain this information what involves longer time needed to detect the intruder.

Table 2. Results of an 90/10 intruders detection in the form of: (number of simulation which an intruder was removed, average time needed to removing of an intruder in constant time periods Δt) with the coincidence of parameters h — number of stored actions and l — length of detectors

-	$h = 16$	$h = 17$	$h = 18$	$h = 19$	$h = 20$...	$h = 27$	$h = 28$	$h = 29$
$l = 4$	(10, 31)	(10, 49)	(8, 58)	(10, 69)	(6, 67)	...	(1, 84.0)	(0, -)	(0, -)
$l = 5$	(10, 19)	(10, 20)	(10, 22)	(10, 28)	(10, 32)	...	(10, 105)	(9, 139)	(9, 120)
$l = 6$	(10, 16)	(10, 15)	(10, 17)	(10, 17)	(10, 18)	...	(10, 32)	(10, 35)	(10, 38)
$l = 7$	(10, 16)	(10, 15)	(10, 16)	(10, 18)	(10, 18)	...	(10, 25)	(10, 26)	(10, 31)
$l = 8$	(10, 16)	(10, 15)	(10, 16)	(10, 17)	(10, 18)	...	(10, 22)	(10, 23)	(10, 27)

In order to obtain more precise evaluation of behavior (small rate of the self-destruction phenomenon) longer detectors ($l = 6$) have to be used. Using detectors that have 6 object representing actions ($l = 6$) involves the necessity of increasing the number of stored actions to 28 or greater. Regarding the time needed to detect an intruder for $l = 6$ it could be stated that optimal size of table of actions is $h = 28$.

5 Heterogeneous Behavior Evaluations

Conclusions gathered in Sect. 4.2 indicate that in order to get precise detection of intruders the number of stored actions has to be extended, which results in longer time of detection of an intruder. In all the tests done in order to show this dependency, the agents have the same parameters (l, h) of behavior evaluation, but in a multi-agent system it is possible to set up different agents with different parameters. This idea is named *heterogeneous behavior evaluations*. Our aim of stating this idea is to use "slow" agents ($l = 6, h = 28$) to restrict the rate of the self-destruction phenomenon and simultaneously to use "rapid" agents ($l = 5, h = 18$) to speed up detection of an intruder. In order to test the idea of heterogeneous behavior evaluations, experiments are done that are analogous to presented in Sect. 4.1 and Sect. 4.2. Obtained results are gathered in Table 3.

Table 3. Results of experiments with heterogeneous acting agents

-	Self-destruction phenomenon	An intruder detection
30% "slow", 70% "rapid"	1.67%	(10, 24)
50% "slow", 50% "rapid"	0.60%	(10, 26)
70% "slow", 30% "rapid"	0.56%	(10, 28)

Results gathered in Table 3 in comparison with the results presented in Table 1 and Table 2 indicate that heterogeneous evaluations are desirable in ethically-social approach to security. Implementation of the proposed idea enables to obtain evaluation possibilities which unite merits of "slow" and "rapid" agents

without manifestation of disadvantages of this two agents' groups. Analyzing results for 50%/50% case the time needed to detect of an intruder is approximate to time of detection for agents with $l = 5$ detectors ("rapid" agents which perform more rapid evaluation of behavior), but the rate of the self-destruction phenomenon is rather specific for agents with $l = 6$ detectors ("slow" agents which perform more precise evaluation of behavior).

6 Conclusion

Dependencies between parameters of behavior evaluation process and the quality of behavior evaluation (rates of intruders detection and the self-destruction phenomenon) have been presented in this paper. The main conclusion is that in order to obtain precise detection great amount of information about agents' behavior have to be used, which involves using long detectors ($l = 6$) and storing great number of actions performed in the past (the size of table of actions $h = 28$). On the other hand, using shorter detectors ($l = 5$) and smaller tables of actions ($h = 18$) causes acceleration of the process of intruder detection, however, this process is not so precise as using longer detectors ($h = 6, l = 28$).

Distributed character of behavior evaluation performed by agents in multi-agent system led us to the idea of heterogeneous behavior evaluations. In this approach different groups of agents in the secured system are set up with different parameters. Performed tests indicate the usefulness of this idea. Obtained possibilities of evaluation due to heterogeneity demonstrate mutual advantages of using short ($l = 5$) and long ($l = 6$) detectors.

References

1. Cetnarowicz, K., Cięciwa, R., Rojek, G. (2005) Behavior Evaluation with Actions' Sampling in Multi-agent System, In Lecture Notes in Artificial Intelligence, Vol. 3690, Springer-Verlag Berlin Heidelberg, 490–499
2. Cetnarowicz, K., Cięciwa, R., Rojek, G. (2005) Behavior Evaluation with Earlier Results Collection in Multi-agent System, In Preprint Proceedings of The Agent Days 2005, Malaga, 7-8 July 2005, 77–84
3. Forrest, S., Perelson, A. S., Allen L., Cherukuri R. (1994) Self-nonsel Self Discrimination in a Computer. In Proc. of the 1994 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos, 202–212
4. Hofmeyr, S. A., Forrest, S. (2002) Architecture for an Artificial Immune System. Evolutionary Computation, vol. 7, no. 1, 45–68
5. Rojek, G., Cięciwa, R., Cetnarowicz, K. (2005) Algorithm of Behavior Evaluation in Multi-agent System, In Lecture Notes in Computer Science, Vol. 3516, Springer-Verlag Berlin Heidelberg, 711–718
6. Schetina E., Green K., Carlson J. (2002) Bezpieczeństwo w sieci. Wydawnictwo HELION, Gliwice

Semi-elitist Evolutionary Multi-agent System for Multiobjective Optimization

Leszek Siwik and Marek Kisiel-Dorohinicki

Department of Computer Science
AGH University of Science and Technology, Kraków, Poland
{siwik, doroh}@agh.edu.pl

Abstract. The paper presents some modification of the idea of an evolutionary multi-agent system for multiobjective optimization, dealing simultaneously with the stagnation of evolutionary process and the loss of agents representing high-quality solutions. The main mechanisms proposed follow the idea of *elitist operators* known from classical evolutionary algorithms, yet in this case the *elite* does not take part in the evolutionary process. Some preliminary results based on a typical multi-objective problem presenting the most important features of the proposed approach are also discussed.

1 Introduction

In the previous papers devoted to the multiobjective optimization using *evolutionary-multi agent systems* (EMAS) there have been presented promising results obtained for different continuous and discrete problems [6, 5]. Moreover, it has been also shown that the EMAS-based approach can be perceived as a very attractive alternative to “classical” evolutionary algorithms for *Pareto-optimization*.

Unfortunately, experimental studies confirm that multiobjective EMAS, as many other *non-elitist* multiobjective evolutionary techniques, during the process of evolution loses a lot of solutions belonging de facto to the final approximation of the Pareto frontier. The introduction of *elitist operators* responsible for ensuring that the best individuals are directly carried over to next generations [2], obviously solves this problem, yet at the same time it augments another one. The EMAS-based approach, like other algorithms utilizing directly the information about the domination relation, suffers from the problem of stagnation, caused by lack of selective pressure in a population consisting—with time—mostly of non-dominated solutions. Of course this is an undesirable effect because it makes the possibility of discovering further solution(s) rather limited. Additionally, in case of Pareto optimization, it results not only in lack of drifting to the target frontier but also in obtaining its very *limited* approximation (in fact rather its fragment(s)).

In the course of this paper some modification of EMAS-based approach dealing simultaneously with the stagnation of evolutionary process and the loss of agents representing high-quality solutions is discussed. The system is called a *semi-elitist* evolutionary multi-agent system (seEMAS), because the main mechanisms proposed follow the idea of *elitist operators* known from classical evolutionary algorithms, yet in this case

the *elite* does not take part in the evolutionary process. As it turns out, the approach can be perceived as a compromise between gathering all non-dominated solutions found by the algorithm in any generation and assuring that the population does not fall into stagnation.

2 Evolutionary and Agent-Based Multiobjective Optimization

For most real-life decision problems a lot of different factors have to be considered, and the decision maker often has to deal with an ambiguous situation: the solutions which optimize one criterion may prove insufficiently good considering the others. In such a situation it may be helpful to use a *domination relation*, which means that one alternative is better than another one considering at least one criterion, and not worst considering the others (so-called *weak domination*). A solution of the multiobjective optimization problem in the *Pareto sense* means determination of all non-dominated alternatives (so-called *Pareto set*) from the set of all feasible solutions.

For the last 20 years a variety of evolutionary multiobjective optimization algorithms (*EMOA*) have been proposed [1, 4]. In Deb's typology among the most important and interesting methods there are distinguished [2]:

- *elitist EMOAs*—i.e. multiobjective evolutionary algorithms based on so-called elite-preserving operators. These operators allow the elites of a population to be directly carried over to the next generation (like in Rudolph's algorithm, distance-based Pareto GA, strength Pareto EA, thermodynamical GA, Pareto-archived evolution strategy, multi-objective messy GA, multi-objective micro GA etc.),
- *non-elitist EMOAs*—i.e. multiobjective evolutionary algorithms which are focused on emphasizing all non-dominated solutions in a population equally and simultaneously on preserving a diverse set of multiple non-dominated solutions without exploiting the mechanism of elitism (e.g. vector evaluated GA, vector-optimized evolution strategy, weight-based GA, random weighted GA, niched-pareto GA, or distributed sharing GA).

The approach of *evolutionary multi-agent systems* is both similar as well as different from classical evolutionary algorithms. The key idea of *EMAS* is the incorporation of evolutionary processes into a multi-agent system at a population level [3]. It means, that besides interaction mechanisms typical for agent-based systems (such as communication) agents are able to *reproduce* (generate new agents) and may *die* (be eliminated from the system). A decisive factor of the agent's activity is its fitness, expressed by the amount of possessed non-renewable resource called *life energy*. Selection is realized in such a way that agents with high energy are more likely to reproduce, whereas a low level of energy increases their possibility of death. In fact, *all* decisions about actions to be performed (including death and reproduction) are made autonomously by agents, and thus *EMAS* may be considered as a computational technique utilizing a *decentralized* model of evolution, unlike classical evolutionary computation. What is more, since agents usually operate in some (virtual) space and their interactions (e.g. selection) are limited to their close neighborhood, this model is also *distributed* like in parallel evolutionary algorithms.

In EMAS-based multiobjective optimization (in the Pareto sense) each agent represents a feasible solution to a given problem. By means of communication agents acquire information, which allows for the determination of the *domination relation* with respect to the others. Then dominated agents transfer a fixed amount of *life energy* to their dominants. This way non-dominated agents (representing successive approximations of the Pareto set) gain more *life energy* and reproduce, while dominated agents die [6].

3 Testing Problem and Comparison Criteria

The experimental results presented in the course of this paper are based on the so-called *MaxEx* multi-objective problem, which is defined as follows:

$$\text{MaxEx} = \begin{cases} f_1(x) = 1.1 - x_1 \\ f_2(x) = 60 - \frac{1+x_2}{x_1} \\ 0.1 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 5 \end{cases}$$

The Pareto frontier and the Pareto set of such two-objective problem can be represented as it is shown in fig. 1. As one may notice, it is a quite simple problem with coherent Pareto frontier and Pareto set as well, which however has some interesting properties (described precisely e.g. in [2]). For the sake of clarity, this target frontier will be omitted in further figures presenting its approximations obtained by the analyzed algorithms.

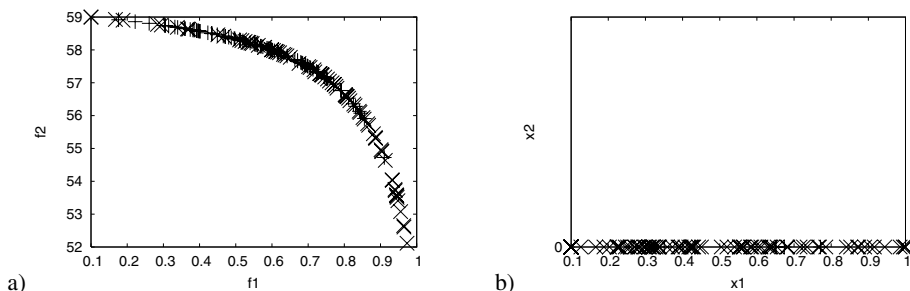


Fig. 1. Visualization of the Pareto frontier (a) and the Pareto set (b) for *MaxEx* problem

In the literature there at least three criteria distinguishing good approximations of Pareto frontiers can be found: the closeness to the target frontier (the closer the better), the number of individuals belonging to the proposed approximation (the more the better), and the dispersing of individuals over the whole frontier [2]. All these criteria will be considered in the discussion below.

4 Towards a Semi-elitist Evolutionary Multi-agent System

Although EMAS-based approach has proved its usefulness for multiobjective optimization, obviously there are still many features that could be improved. First of all, especially in case of simple problems, EMAS population sometimes falls into stagnation. In

fig. 2a there are presented: the number of individuals belonging to the Pareto frontier and the size of EMAS population in consecutive steps during solving *MaxEx* problem. As one may see, after c.a. 500 steps of evolution the size of the population stabilizes. There are two possibilities: either the number of agents with very low *life energy* level (i.e. dying agents) is the same as the number of agents with very high level of resources (i.e. reproducing agents), or there are no agents with very high or very low level of *life energy*, and in consequence there are no reproducing or dying agents—which was confirmed during the experiments. This phenomenon may be explained analyzing the number of individuals belonging to the Pareto set. As it is confirmed in fig. 2a, with time, the whole population consists of only *nondominated* individuals, and during agents' meetings the condition responsible for the flow of energy (cf. section 2) is not fulfilled. In consequence, the transfer of energy is lower and lower, until it disappears entirely.

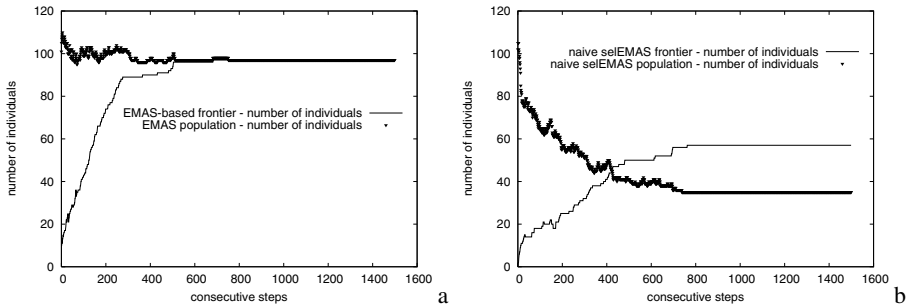


Fig. 2. Size of population and size of Pareto set for *MaxEx* problem obtained by a) EMAS- and b) 'naive' seEMAS-based algorithms in consecutive steps

Another feature of EMAS-based approach, undesirable especially in case of multiobjective optimization—but which is de facto a natural feature of every optimization technique based on the evolution principle—is that during the evolution process some valuable solutions can be (and are in fact) irreparably lost. In singleobjective optimization it is not so dangerous if still the algorithm is able to find at last the global optimum—however in case of multiobjective optimization—because in most cases the algorithm should find as many nondominated solutions as it is possible—some mechanisms responsible for *storing* any found nondominated alternatives (of course as long as they are still nondominated) could be introduced. Fig. 2b presents the size of the population and the number of individuals belonging to the Pareto frontier for the first, *naive* modification of EMAS. In this case an additional, special island (so-called *elitist* island) was introduced to the system. The best individuals (that can be perceived as the *elite* of the agents' society) were to migrate to this island. Because there are only coming in paths to the elitist island, such mechanism should solve the problem with losing valuable alternatives during the process of evolution.

Of course the question appears, which agents can migrate to the elitist island. One of at least several possibilities, consists in utilizing the information about domination relation discovered during the meetings. It may be assumed that the agents that prove

better than at least n other agents constitute the elite, and can migrate to the elitist island. Unfortunately, such approach does not guarantee a proper behavior of the system. As it is shown in fig. 2b, migration of individuals from an *ordinary* island to the elitist one causes dying out the ordinary population. As one may see, starting from 100 individuals the size of population drops down and stabilizes at a level of 40 evolving agents. Even though the system is able to find more nondominated solutions than the population size, there is still the problem with disappearing life energy transfer. In consequence, even faster than in the previous case, all (or almost all) members of population become non-dominated and there are no dying or reproducing agents—so, the process of evolution de facto does not take place. To solve this problem the further modification has been proposed: in place of each migrating elitist agent another one is created, but to avoid the premature convergence of the population on the one hand but, simultaneously, to follow to the direction determined by migrating elitist agent on the other hand—the created individual is its mutated clone.

5 Experimental Studies

In fig. 3 there are presented subsequent approximations of the Pareto frontier for our testing *MaxEx* problem obtained by EMAS- and seEMAS (i.e. semi-elitist evolutionary multi-agent system)-based optimization after 1, 30, 500 and 1500 steps.

As one may see all the frontiers obtained by both approaches are very similar, both initially (after 1 and 30 steps), but also after 1000 and 1500 steps, if the distance to the target frontier and dispersing over the whole frontier are considered. However, if the number of found nondominated individuals is considered, initially EMAS-based technique is better but with time seEMAS-based frontier is much more numerous. It seems thus that introducing *semi-elitism* into EMAS, allows for obtaining frontiers, which are not worse regarding the closeness to the target frontier and even dispersing of nondominated individuals over the whole frontier and is strongly better regarding the number of found Pareto-optimal solutions.

Of course it is difficult to reliably compare the considered approaches relying on charts representing consecutive approximations of the Pareto frontier. That is why in fig. 4 there are presented some characteristics allowing for deeper analysis of the obtained results. In fig. 4a the number of individuals belonging to the obtained EMAS- and seEMAS-based Pareto frontiers in consecutive steps of the system run are presented. Initially, there are more individuals belonging to the Pareto frontier in the case of EMAS-based optimization. However after ca. 500 steps seEMAS-based frontier becomes more numerous than the EMAS-based one. Additionally in fig. 4a the number of individuals belonging to the EMAS and seEMAS population are also presented. Because (without any additional mechanisms) these approaches do not guarantee a fixed size of the population, it could occur that the increasing number of individuals belonging to the Pareto frontier results from increasing of the population size, however the presented characteristic shows that it is not true. Moreover, as one may see, seEMAS-based population is not as numerous as the EMAS-based one, and nevertheless seEMAS is able to obtain more numerous frontier.

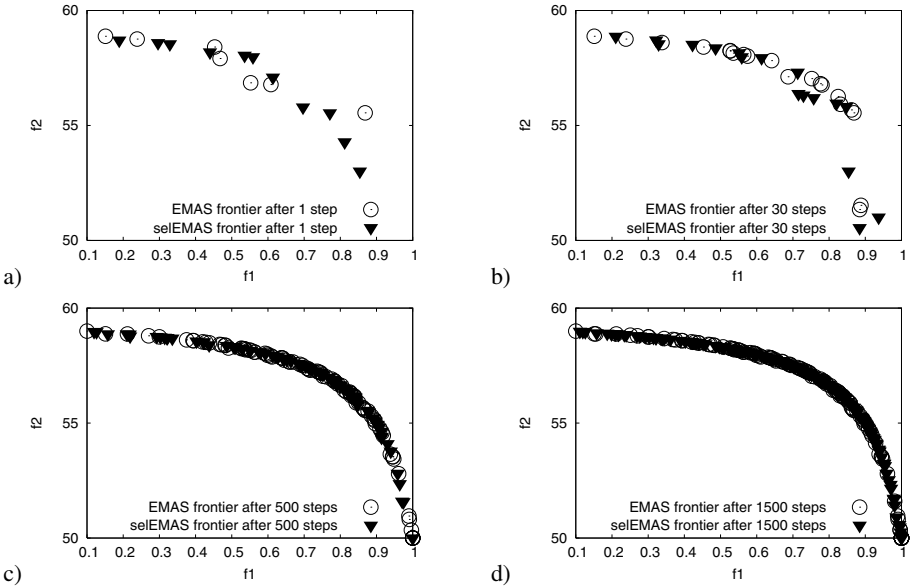


Fig. 3. Pareto frontier approximations for *MaxEx* problem obtained by EMAS and selEMAS after a) 1, b) 30, c) 500 and d) 1500 steps

In fig. 4b there is presented another very important profile i.e. the distance to the target solution. As one may see, this chart confirms which was observed in fig. 3—the distance to the target Pareto frontier obtained by both algorithms is very similar.

Figures 4c and 4d show the diversity of individuals belonging to the obtained Pareto sets. The especially important in this context characteristic is presented in fig. 4c, because the diversity in the space of x_1 decision variable ensures (in the case of *MaxEx* problem) dispersing individuals over the whole Pareto set. So, analyzing these charts, it seems that EMAS focuses firstly on drifting to the target frontier, and during this process diversity is being gradually improved, whereas selEMAS keeps the diversity on almost the same level from the very beginning until the end of evolution process—which of course seems to be more desirable especially in such situations when the process of evolutions does not last 1500 steps but is much shorter.

It is worth in this place to analyze the influence of introduced *semi-elitist* mechanisms not only on the diversity of the obtained Pareto set but also on diversity of the whole population. Such profiles are presented in fig. 5a and fig. 5b. Of course because of specific features of *MaxEx* problem the more important is the diversity in the x_1 decision variable space. And as one may see, starting from the same level of the diversity, selEMAS population almost immediately becomes much more dispersed and this remains until the end of the evolution process.

At last one may ask the question—what is the influence of introduced mechanisms if the *real* computation time is considered. Yet it occurs that selEMAS-based optimization

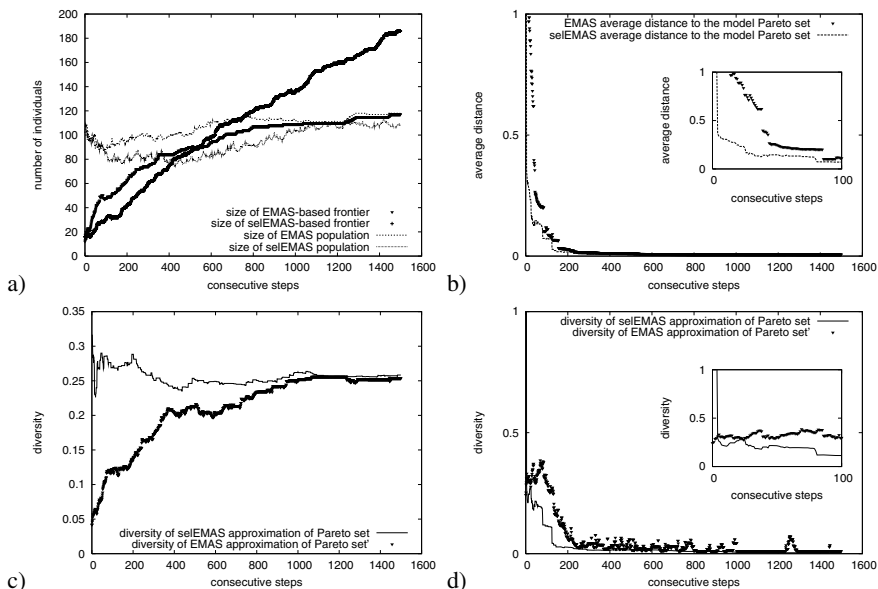


Fig. 4. Characteristics of obtained seEMAS-based Pareto frontier approximations: the number of solutions in the frontiers (a), average distance to the model frontier (b), the diversity of solutions in the frontier for x_1 (c) and x_2 (d) decision variable

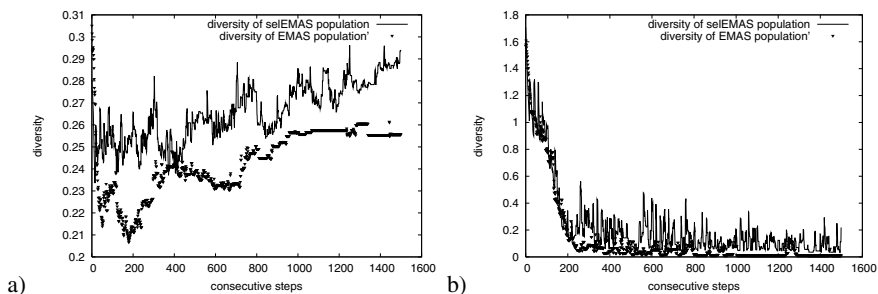


Fig. 5. Further characteristics of obtained Pareto frontier approximations: the diversity of populations for x_1 (a) and x_2 (b) decision variable

in spite of the introduced additional mechanisms is not slower, it seems to be even faster than the EMAS-based one. Performing 1500 steps of evolution starting from the population of 100 agents solving *MaxEx* problem took c.a. 300ms for EMAS and c.a. 220ms for seEMAS¹. Why in spite of the additional mechanisms seEMAS is faster in absolute time horizon? The answer brings fig. 4a: because during almost the whole evolution process the size of seEMAS population is significantly lower (by c.a. 20%) than the size of EMAS population.

¹ Computations have been performed on PC with 512MB RAM and PIII 800MHz processor.

6 Concluding Remarks

Recapitulating, it can be said that introducing *semi-elitism* into EMAS allows—in case of *MaxEx* problem—for obtaining Pareto sets that is not worse than the EMAS-based ones if the distance to the model frontier is considered and that is even significantly better if other characteristics are considered.

Further research will focus on the deeper analysis of the elitist operators for EMAS for various testing problems. Interesting results may be obtained when the elite will be given an opportunity to influence the evolution process on ordinary islands, thus a shift from *semi-elitism* to *elitism* is foreseen to be the important step ahead in the development of the idea of multiobjective EMAS.

References

1. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.
2. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
3. M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In W. I. Grosky and F. Plasil, editors, *SofSem 2002: Theory and Practice of Informatics*, volume 2540 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
4. A. Osyczka. *Evolutionary Algorithms for Single and Multicriteria Design Optimization*. Physica Verlag, 2002.
5. L. Siwik and M. Kisiel-Dorohinicki. Balancing of production lines – evolutionary agent-based approach. In G. Lefranc, editor, *Management and Control of Production and Logistics — MCPL 2004*, pages 319–324, 2004.
6. K. Socha and M. Kisiel-Dorohinicki. Agent-based evolutionary multiobjective optimisation. In *Proc. of the 2002 Congress on Evolutionary Computation*. IEEE, 2002.

Agent-Based Evolutionary Model for Knowledge Acquisition in Dynamical Environments

Wojciech Froelich¹, Marek Kisiel-Dorohinicki², and Edward Nawarecki²

¹ Institute of Computer Science
Silesian University, Sosnowiec, Poland
froelich@konto.pl

² Institute of Computer Science
AGH University of Science and Technology, Kraków, Poland
{doroh, nawar}@agh.edu.pl

Abstract. The basic idea of the approach proposed in this paper is to apply multi-agent paradigm in order to enable the integration and co-operation of different knowledge acquisition and representation techniques. The effective operation of learning process is achieved by evolutionary optimization running at the level of agents' population. In the discussed variant of the model, each agent uses reinforcement learning, and the obtained knowledge is represented as the set of simple decision rules. The approach is illustrated by a particular realization of the system dedicated to the evasive maneuvers problem, together with preliminary experimental results.

1 Introduction

The research in the domain of artificial intelligence has led to the formation of many methods that permit to solve effectively different complex tasks, also in the field of knowledge acquisition. Yet, for many difficult problems, and particularly in case of dynamically changing environments or incomplete and uncertain data, it may be necessary to adjust the method configuration to each specific variant of the task. This leads to changes in the composition and parameters of the algorithm being used (optimization through experimental evaluation). One of possible solutions is the use of hybrid methods, hierarchical composition of algorithms, and also multi-starting methods. Such approach is inspired by the observation of the natural world, where knowledge acquisition processes are realized on different levels of abstraction and the knowledge is represented by different structures (neural networks, evolutionary processes).

The basic idea of the proposed approach is to apply a multi-agent system [1, 8] in order to enable the integration and co-operation of various methods of knowledge acquisition and representation [6]. In the proposed model, the environment represents the problem being solved, and at the same time, enables the mutual agents' observation. Every agent represents a system that learns to achieve complete or partial solution of the considered problem. It is assumed that the agent is situated in the environment and interacts with it, for instance by experimentation with different behavioral strategies. In the environment there can be distinguished objects that are the source of information for learning performed by single agents and the whole population. The task that the every

agent has to perform consists in the identification of the features of objects, relations between them, or even in specifying the effective strategy of objects' control. The objects' characteristics may change during the learning process, also due to the (control) signals obtained from the system. Thus in fact the knowledge acquisition is related to the constant mutual interaction between the environment and system [2].

The issues being considered in this paper apply to the case, when the environment, from which the knowledge is to be derived, is not entirely known. In such a case, the knowledge acquisition takes place by the realization of the sequence of agent's decisions, that by intention are to lead to reaching the goal, while their intermediate effects (environment state changes) give information about its characteristics. So, even if the goal is not reached, agent's activities will give the information resulting from the environment response.

In the following sections the sketch of the model is presented. Then the details of the knowledge acquisition process are discussed. These general considerations are illustrated by a sample realization of the system dedicated for solving the evasive maneuvers problem. Preliminary experimental results conclude the work.

2 Knowledge and Learning in a Multi-agent System

Multi-agent system is composed of the set of agents working in a common environment. We can say about dual nature of such defined system, in which on one hand we have to do with the environment that surrounds the agents, and on the other hand with the population of agents building the knowledge representation, which applies to this environment. Thus in a given moment of time, the multi-agent system state is represented by the 2-tuple:

$$MAS \equiv \langle ES, AG \rangle, \quad (1)$$

where: ES represents the environment state, and AG denotes the set of agents (their states).

In the proposed model, the environment can be observed by agents through signals, and in fact the environment state is represented only by the set of signals ($es \in ES$). The signals can be interpreted as *a priori* unknown, but measurable quantities representing phenomena occurred in it.

The initial agents' population is generated on the strength of the expert knowledge and is equipped with a random set of features. Then the operation of this system is considered in consecutive moments of time. The set of environment signals is the source of information for all agents. Those signals are received by sensors, each agent is equipped with (observation), and the agent undertakes some activities. Agent's activities can mean the realization of single actions or the sequence of them. In consequence the changes of observed signals can appear, which in turn may be observed by the agents in future. The reactive method of agents' activity is assumed, which means, that agents do not perform mathematical calculations and algorithmically controlled analysis of the history of their observations and actions. Planning in the meaning of building and estimation of behavioral patterns is neither realized.

The knowledge acquisition is realized by using reinforcement learning [2] paradigms and a multi-agent evolution [5]. Reinforcement learning leads to the creation and

selection of decision rules that allow for reaching the agent's goal. The multi-agent evolution is to improve the effectiveness of the learning process through optimization of individual agent's features, which can apply among others to its observation abilities (e.g. sensors sensitivity on the environment signals change), and may be the parameters of the decision process. The agents are equipped with knowledge that can be classified according to its nature as a-priori, delivered by the designer, or a-posteriori acquired during the learning process. The a-posteriori knowledge learned by every agent can be considered as:

- phylogenetic knowledge (genotypic characteristics obtained through the
- multi-agent evolution process), ontogenetic knowledge (acquired through the adaptation in the environment).

3 The Structure of a Learning Agent

The agent's internal structure has been defined *a priori* at the stage of model design [2], and is composed of the following functional blocks. By means of **sensors**, agent observes the environment signals. Sensors are characterized among others by sensitivity that describes the minimal change of the environment signals that are being recorded. **Iconic memory** stores successive observations and decisions. In case of incomplete environment observation, an agent may lack sufficient information necessary to make an appropriate decision. This short term memory provides some historical data, which may be used to discover the dynamics of the environment changes. With the use of **effectors**, the agent puts into practice the activities in the environment. Agent activities can be either single reactions (actions) to received stimulus or sequences of actions leading to particular goals. **Knowledge base** is composed of decision rules, which consists of some observation pattern (premise) and assigned agent's decision (conclusion). **Genotype** describes selected elements of the agent's structure, its sensory and morphological features. Examples of the agent's genotype features can be: sensors sensitivity or environment observation range. **Energy** is a consumable and renewable resource that describes agent's usability in the realization of the task. **Decision algorithms** constitute the agent's structure kernel, and make it possible to take decisions and implementation of residual functions connected with agent's activity.

The the actual agent's state is represented by the 6-tuple:

$$ag \equiv \langle s, d, m, e, g, R \rangle \quad (2)$$

where: s denotes observation vector, d – decision vector, m – iconic memory matrix, e – agent's energy, g – genotype vector (phylogenetic knowledge), R – (ontogenetic) knowledge base.

Figure 1 illustrates a single step of the agent's activity. The signals observed from the environment are stored in the observation vector. A knowledge base should provide a correlation of the observation vector with an adequate decision leading to accomplishing the task. Due to the limited volume of memory, an agent cannot remember all observations, and store in the knowledge base only patterns that represent subsets of

the observation space. An agent makes classification (matching) of the obtained observation vector in a set of observation patterns. As a result the related agent's decision is determined, on the basis of which agent realizes actions in the environment. The observation and decision vectors are then stored in the iconic memory.

The (ontogenetic) knowledge base of an agent consists of simple decision rules (classifiers): $r \equiv \langle w, a, h \rangle \in R$, consisting of condition part w ($w_i \in \mathbf{R}$), decision part a ($a_i \in \mathbf{N}$), and auxiliary attributes h (performance weight, frequency of activations, actual policy gauge). On the basis of the auxiliary attributes the rules can be filtered out after some periods of learning. For example the classifiers which are less frequent or possess lower performance can be deleted from the knowledge base. Thus the amount of classifiers can be adopted to the requirements of the particular task of an agent.

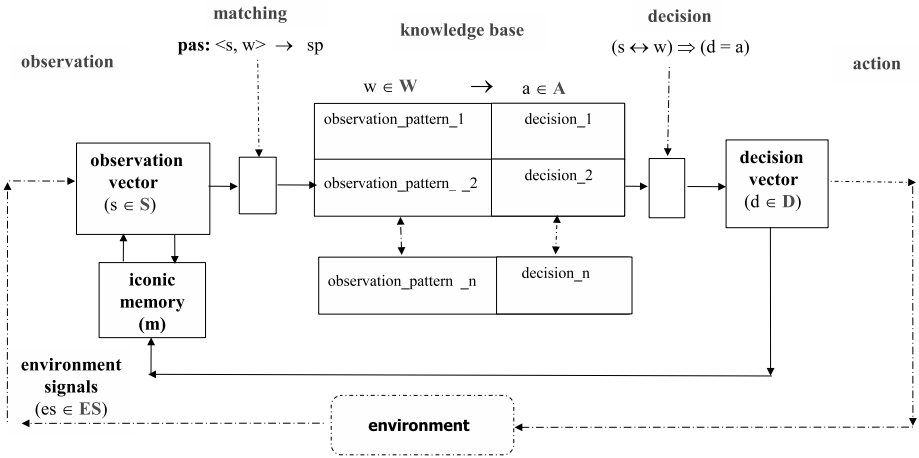
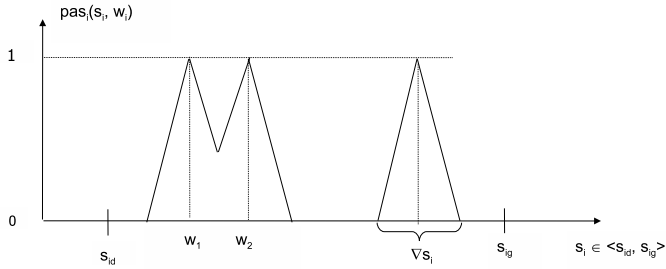


Fig. 1. Agent's activity diagram

4 Knowledge Acquisition Process

The agent's decision process is in fact divided into particular steps that it undertakes in order to achieve its goal, which is the reward received from the environment (reinforcement learning). The value of rewarding function is usually known after finishing a total decision sequence in a defined time horizon. In each step an agent observes the environment and gets vector $s \in S$, then it selects action $a \in A$. The environment responds in a way that after finishing the actions by all agents of the population (a learning episode), it informs every agent about the reward granted. Learning consists in finding the optimal strategy $S \rightarrow A$ where S is the set of environment conditions, A – the set of agents' activities. Considering the sequence of actions, the learning means searching for strategy $S \times A \rightarrow A$.

Let us assume that $pas : S \times W \rightarrow \mathbf{R}$ is the matching function and $s_p \in \mathbf{R}$ is the matching threshold. If the condition $pas(s, w) \geq s_p$ is satisfied, it is assumed that the considered rule is a candidate to activate (fire), which means that its decision part a is a candidate to fulfill the agent's decision vector d . In general, the function $pas(s, w)$



$$pas_i(s_i, w_i) = \begin{cases} \frac{\nabla s_i - r}{\nabla s_i} & , \quad \text{for } r \leq \nabla s_i \\ 0 & , \quad \text{for } r > \nabla s_i \end{cases} \quad \text{where: } \begin{cases} \nabla s_i - \text{parameter stored in genotype,} \\ r = |w_i - s_i| \end{cases}$$

Fig. 2. Matching function

can be considered in n -dimensional observation space S . For the simplification of the computational complexity, it was assumed that $pas(s, w) = \sum_{i=1}^n w s_i \cdot pas_i(s_i, w_i)$, where s_i, w_i are the components of vectors s and w respectively (fig. 2). The value of $w s_i$ is the weight reflecting the importance of the i -th component of the observation vector for the decision process and for the purpose of the following experiments has been encoded into the agent’s genotype vector.

During creation of a new agent, its **ontogenetic knowledge** does not exist (its knowledge base is empty). Then, an agent makes decisions at random, which applies to the way of activity (exploration or exploitation).

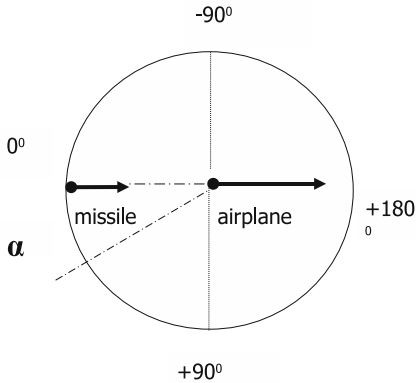
1. In case it choses the exploration, a new rule in the knowledge base is generated. Its conditional part is the observation pattern (obtained on the basis of the observation vector). Decision part is generated at random. A new rule must be unique in the knowledge base.
2. Second possibility is the choice of knowledge exploitation. The agent tries to classify the observation vector according to the previously stored observations. Every unclassified observation generates a new classifier in it’s knoledge base. In this case, the obtained observation vector is the subject to matching in the knowledge base through comparisons with patterns in the conditional parts of rules already existing in the knowledge base.
 - If there is a lack of matching pattern, the complement of the knowledge base is undertaken and a new rule is added just like in the case of exploration.
 - In case of correct matching of more than one pattern in the knowledge base, decision is undertaken at random (the roulette wheel method has been used), the probability of choosing the particular rule depends on its performance weight attribute (the adequate auxiliary attribute of a rule).
3. The final effect of the decision process is the choice of the rule (activation) in the knowledge base. The selected rule is then marked by modification of the value of the usability indicator (another auxiliary attribute). The attribute specifying the number of the rule’s activation is also updated.

4. The agent executes actions as a result of the decision process. The execution of an action by the agent involves spending the energy resource. The decision vector can be stored in the iconic memory.
5. If the agent gains the energetic reward, it modifies the performance weight for all used rules (that led to the success). In order to optimize the parameters of the knowledge base (size, searching speed), the rules with the low performance weight and rarely used (the values of related auxiliary attributes below certain threshold) are removed from the knowledge base.

The acquisition of **phylogentetic knowledge** is realized by the evolutionary processes. Genotype optimization takes place at the population level and takes advantages of the multi-agent evolution paradigm [5]. In this model classical selection mechanism realized on the basis of global fitness function cannot be used, since it leads to the introduction of the order into the phenotype space, which is not always adequate with the space structure of the given problem's solutions. Concerning the above, the energetic selection method observed in biology [7] has been introduced. Every agent of the initial population is provided with a random allocation of the resource, called *life energy*. Having undertaken actions, the agents spend the energy, thus lose some part of energy possessed (energetic cost). Agents, which energy drops below a certain level, are to be eliminated. Agents, which achieve success during the realization of a task, get the energetic reward, which allows for surviving in the environment. Agents, which reach certain energetic level may undergo reproduction. The agent's genotype is the subject of mutation during reproduction. In this way, it is possible to gradually improve the efficiency of agents in the population.

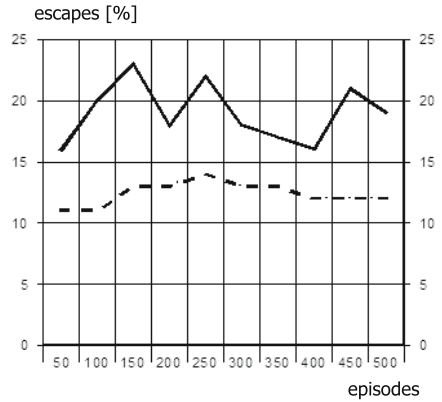
5 Experimental Results

The method of learning an agent described above has been applied to the evasive maneuvers problem (pursue and evasion game) in a two-dimensional space for a single pursuer (rocket) and a single evasive object (plane). In this case, towards the flying plane, the rocket is launched, which goal is to hit the plane. The rocket is led automatically (in a deterministic way) at the target. The plane should be capable of evading the racket by changing the parameters of flight, i.e. the direction and speed. The plane is equipped with measuring devices that inform about the parameters of the approaching rocket flight: the distance, mutual flight angle, rocket's speed. On the basis of the radar readings the agent controlling the plane should be capable of learning how to change the direction and speed of the plane to avoid the hit. Agent learning task has been divided into learning episodes. One of the initial conditions of each episode was the mutual location of the objects: the rocket and the plane (fig. 3a). The episode ends when the plane is hit or manages to escape. The degree of difficulty has been established (by changing dynamical parameters of the objects) in such a way that the trivial strategies, consisting in continuous turning of the plane in one direction, did not allow to escape. The reference point for executed experiments was the case of using by the agent the random strategy of controlling the plane's escape (consisting in random changes of flight direction and speed), which is illustrated in fig. 3b. In this case, the number of escapes achieved, did not exceed 40% of all learning episodes.



$\alpha \in <-90^\circ, +90^\circ>$,
 $\alpha = k \times 180$, $k \in <-5, +5>$

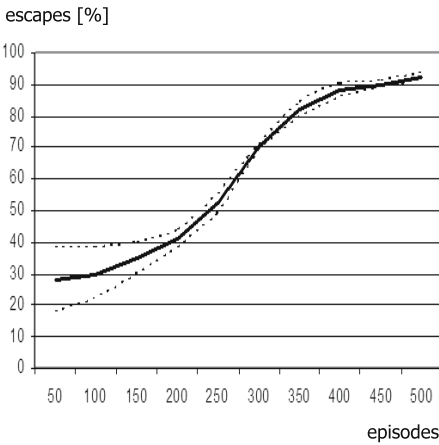
a) Initial position of pursuer and evader



————— The best agent
 - - - - - Average of the population

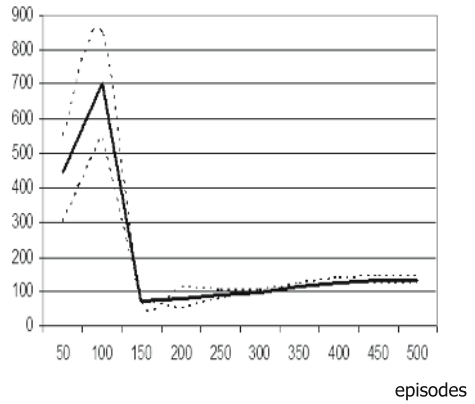
b) Random strategy of the evader

Fig. 3. Preliminary conditions for the simulation



————— the best agent
 - - - - - standard deviation

a) The percentage of escapes during learning process



————— the best agent
 - - - - - standard deviation

b) The amount of rules stored by the best agent

Fig. 4. Effectiveness of the best agent

Further, the tests had been executed, during which, the agent’s learning based on the suggested model had been performed. Obtained results have shown the effectiveness of the learning method being used. The learning effectiveness for the best population agent, let to achieve over 90% ratio of the plane escapes from among all learning episodes (fig. 4a). The effectiveness of the best agent is comparable to the results

achieved in other systems [4, 3]. The optimization of the agent's knowledge base size also has been achieved, i.e. the reduction of the number of decision rules (fig. 4b).

6 Concluding Remarks

The main idea of the approach was based on the agent paradigm in order to organize the hybrid method of knowledge representation and acquisition [6], together with the evolutionary optimization of the population of co-operating algorithms—agents [5]. A suitable model has been elaborated, supporting analysis of the learning processes that occur on different levels of abstraction.

Based on the presented realization and performed experiments, it may be said that the suggested idea fulfills requirements applying to the effectiveness of the obtained solutions. It is also foreseen to have great universality within the chosen range of applications. Further improvements of the suggested knowledge acquisition model require more experiments concerning other applications, which will be the subject of continued research.

References

1. J. Ferber. *Multi-Agent Systems*. Addison-Wesley, 1999.
2. W. Froelich. Evolutionary multi-agent model for knowledge acquisition. In *Intelligent Information Processing and Web Mining (IIPWM'05)*, Advances In Soft Computing. Springer, 2005.
3. D. Gordon and D. Subramanian. A multistartegy learning scheme for assimilating advice in embedded agents. In *Proc. of the Second Intl. Workshop on Multistrategy Learning*, 1993.
4. J. Grefenstette, C. Ramsey, and A. Schultz. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4), 1990.
5. M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In W. I. Grosky and F. Plasil, editors, *SofSem 2002: Theory and Practice of Informatics, Lecture Notes in Computer Science*. Springer-Verlag, 2002.
6. M. Kisiel-Dorohinicki, G. Dobrowolski, and E. Nawarecki. Agent populations as computational intelligence. In L. Rutkowski and J. Kacprzyk, editors, *Neural Networks and Soft Computing*, Advances in Soft Computing. Physica-Verlag, 2003.
7. S. Wierzchoń. Multimodal optimization with artificial immune systems. In *Intelligent Information Systems*. Physica-Verlag, 2001.
8. M. Wooldridge, editor. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

Quantum-Behaved Particle Swarm Optimization Algorithm with Controlled Diversity

Jun Sun, Wenbo Xu, and Wei Fang

Center of Intelligent and High Performance Computing,
School of Information Technology, Southern Yangtze University,
No. 1800, Lihudadao Road, Wuxi Jiangsu 214122, China
{sunjun_wx, xwb_sytu, wxfangwei}@hotmail.com

Abstract. Premature convergence, the major problem that confronts evolutionary algorithms, is also encountered with the Particle Swarm Optimization (PSO) algorithm. In the previous work [11], [12], [13], the Quantum-behaved Particle Swarm (QPSO) is proposed. This novel algorithm is a global-convergence-guaranteed and has a better search ability than the original PSO. But like other evolutionary optimization technique, premature in the QPSO is also inevitable. In this paper, we propose a method of controlling the diversity to enable particles to escape the sub-optima more easily. Before describing the new method, we first introduce the origin and development of the PSO and QPSO. The Diversity-Controlled QPSO, along with the PSO and QPSO is tested on several benchmark functions for performance comparison. The experiment results testify that the DCQPSO outperforms the PSO and QPSO.

1 Introduction

The Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart [5], is a stochastic optimization that can be likened to the behavior of a flock of birds. It has been used to solve a range of optimization problems, including neural network training [6] and function minimization. Since its origin in 1995, several attempts have been made to improve the performance of the original PSO ([1], [2], [4], [8], [9], [10], [17]). The original PSO, however, is not a global optimization algorithm, as has been demonstrated by F. van den Bergh in [3]. In our previous work [11], [12], [13], we proposed a novel global-convergence-guaranteed PSO, Quantum-behaved Particle Swarm Optimization (QPSO).

Like other evolutionary algorithm, the PSO as well as QPSO, confront the problem of premature convergence, which results in great performance loss and sub-optimal solutions. In the PSO or QPSO, the fast information flow between particles seems to be the reason for clustering of particles. Diversity declines rapidly, leaving the PSO or QPSO algorithm leads to low diversity with fitness stagnation as an overall result. In this paper, we propose a Diversity-Controlled Quantum-behaved Particle Swarm Optimization (DCQPSO). In the DCQPSO, a threshold value was set for population's diversity measure to prevent premature convergence and therefore enhance the overall performance of the QPSO. The rest part of the paper is organized as follows. In the

next section, the PSO is introduced. The origin and development of QPSO is described in Section 3 and the DCQPSOs are proposed in Section 4. Section 5 is the experiment results and discussion. Some conclusion remarks are given in Section 6.

2 PSO Algorithms

The PSO is a population-based optimization technique, where a population is called a swarm. Each particle represents a possible solution to the optimization task at hand. During each iteration each particle accelerates in the direction of its own personal best solution found so far, as well as in the direction of the global best position discovered so far by any of the particles in the swarm. Let M denote the swarm size. Each individual i ($1 \leq i \leq M$) has the following attributes: A current position in the search space $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$, a current velocity $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,n})$, and a personal best position in the search space $P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,n})$. During each iteration, each particle in the swarm is updated using (1) and (2). Assuming that the function f is to be minimized, that the swarm consists of n particles, and that $r_1 \sim U(0,1)$, $r_2 \sim U(0,1)$ are elements from two uniform random sequences in the range $(0,1)$, then

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_{1,i}(t) \cdot [p_{i,j}(t) - x_{i,j}(t)] + c_2 \cdot r_{2,i}(t) \cdot [p_{g,j}(t) - x_{i,j}(t)] \tag{1}$$

for all $j \in 1,2,\dots,n$, thus, $v_{i,j}$ is the velocity of the j th component of the velocity of the i th particle, and c_1 and c_2 denote the acceleration coefficients. P_g is the global best position found by any particle during all previous steps. The new position of a particle is calculated using

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

The value of each component in every V_i vector can be clamped to the range $[-v_{\max}, v_{\max}]$ to reduce the likelihood of particles leaving the search space. The variable w in (1) is called the inertia weight, whose value is typically setup to vary linearly from 1 to near 0 during the course of training run. Acceleration coefficients c_1 and c_2 also control how far a particle will move in a single iteration.

3 Quantum-Behaved Particle Swarm Optimization

Trajectory analyses in [5] demonstrated that, to guarantee convergence of the PSO algorithm, each particle must converge to its local attractor $q = (q_1, q_2, \dots, q_D)$, of which the coordinates are:

$$q_d = (c_1 r_1 p_{id} + c_2 r_2 p_{gd}) / (c_1 r_1 + c_2 r_2), (d=1,2,\dots,n) \tag{3}$$

or

$$q_d = \varphi \cdot p_{id} + (1 - \varphi) \cdot p_{gd}, \varphi \sim U(0,1), (d=1,2,\dots,n) \tag{4}$$

Assume that there is one-dimensional Delta potential well on each dimension at point q and each particle has quantum behavior. For simplicity, we consider a particle in one-dimensional space, with point q the center of potential well. Solving the *Schrödinger equation*, we can get the normalized the following probability distribution function F

$$F(y) = \int_{-\infty}^y Q(y) dy = e^{-2|q-x|/L} \tag{5}$$

where L determines search scope of each particle. Employing Monte Carlo method, we can obtain the position of the particle

$$x = q \pm \frac{L}{2} \ln(1/u) \quad u = rand(0,1) \tag{6}$$

where u is a random number uniformly distributed in $(0, 1)$.

A global point called Mainstream Thought or Mean Best Position of the population is introduced into PSO. The global point, denoted as $mbest$, is defined as the mean of the $pbest$ positions of all particles. That is

$$mbest = \frac{1}{M} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M p_{i1}, \frac{1}{M} \sum_{i=1}^M p_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M p_{id} \right) \tag{7}$$

where M is the population size and P_i is the $pbest$ position of particle i . Then the value of L and the position are evaluated by

$$L = 2 \beta \cdot |mbest - x| \tag{8}$$

$$x := q \pm \beta \cdot |mbest - x| \cdot \ln(1/u) \tag{9}$$

where parameter β is called Contraction-Expansion Coefficient, which could be tuned to control the convergence speed of the algorithms. The PSO with equation (19) is called Quantum-behaved Particle Swarm Optimization (QPSO).

4 Diversity-Controlled QPSO

As we know, a major problem with PSO and other evolutionary algorithms in multi-modal optimization is premature convergence, which results in great performance loss and sub-optimal solutions. In a PSO system, with the fast information flow between particles due to its collectiveness, diversity of the particle swarm declines rapidly, leaving the PSO algorithm with great difficulties of escaping local optima. Therefore, the collectiveness of particles leads to low diversity with fitness stagnation as an overall result. In QPSO, although the search space of an individual particle at each iteration is the whole feasible solution space of the problem, diversity loss of the whole population is also inevitable due to the collectiveness.

In 2002, Ursem has proposed a model called Diversity-Guided Evolutionary Algorithm (DGEA) [14], which applies diversity-decreasing operators (selection,

recombination) and diversity-increasing operators (mutation) to alternate between two modes based on a distance-to-average-point measure. The performance of the DGEA clearly shows its potential in multi-modal optimization.

Also in 2002, Riget *et al* [15] adopted the idea from Usrem into the basic PSO model with the decreasing and increasing diversity operators used to control the population. This modified model of PSO uses a diversity measure to have the algorithm alternate between exploring and exploiting behavior. They introduced two phases: attraction and repulsion and the swarm alternate between these phases according to its diversity. The improved PSO algorithm is called Attraction and Repulsion PSO (ARPSO) algorithm.

Inspired by works undertaken by Usrem and Riget *et al*, we introduce the Diversity-Controlled model in Quantum-behaved PSO. The diversity is measure by the following formula.

$$diversity(S) = \frac{1}{|S| \cdot |A|} \cdot \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^D (x_{ij} - \bar{x}_j)^2} \tag{10}$$

where S is the swarm, $|S|=M$ is the population size, $|A|$ is the length of longest the diagonal in the search space, D is the dimensionality of the problem, x_{ij} is the j th value of the i th particle and \bar{x}_j is the j th value of the average point.

But unlike to their works, we only set a low bound to the diversity of the population. The procedure of the algorithm is as follows. After initialization, the algorithm is running in attraction mode that guaranteed by setting the value of β smaller than 1.0. In [13], using stochastic simulation, we found out that the particle will converge when $\beta < 1.778$ and otherwise will diverge. In this paper, the attraction mode is realized by varying β from 1.0 to 0.5 over the running. That is,

$$\beta = (1.0 - 0.5) \times (MAXITER - t) / MAXITER - 0.5 \tag{11}$$

where $MAXITER$ is the maximum number of iterations and t is the number of current iteration. In the course of the running of the algorithm, if the diversity measure of the population declines to below the threshold value d_{low} , the particles will explode to increase the diversity until it is larger than d_{low} . There are two method of making the particle to explode. One method is to control the Contraction-Expansion Coefficient β . That is, we can set $\beta = \beta_0$ ($\beta_0 > 1.778$) once the diversity is lower than the threshold value d_{low} . In this paper the DCQPSO using this method is called DCQPSO1. The other method of increasing the diversity is initializing the Mean Best Position ($mbest$) of the population across the search space once the diversity is smaller than d_{low} . The reason for the initialization of $mbest$ is that when the diversity is low, the distance between the particle and the $mbest$ position is too small for the particle to escape the local optima as can be seen from equation (9). Therefore initializing the $mbest$ could enlarge the gaps between particles and the $mbest$ position, consequently making particles explode temporarily. The DCQPSO with this operation is called DCQPSO2. The DCQPSO algorithms are formulated as follows.

```

Initialize the population
for t=1 to MAXITER
  Compute the mbest of the population;
  measure the diversity of the population by (10)
   $\beta = (1.0 - 0.5) * (\text{MAXITER} - t) / \text{MAXITER} + 0.5$ ; (in attraction
mode)
  if (diversity < dlow) (in explosion mode)
     $\beta = \beta 0$ ; (for DCQPSO1)
    (or initializing the mbest; (for DCQPSO2));
  endif
  for i=1 to population size M;
    if  $f(P_i) < f(X_i)$  then  $X_i = P_i$ 
    pg = argmin( $P_i$ );
    for d=1 to dimension n
       $f_i = \text{rand}(0, 1)$ ;
       $q = f_i * p_{id} + (1 - f_i) * pg_d$ ;
       $u = \text{rand}(0, 1)$ ;
      if  $\text{rand}(0, 1) > 0.5$ ;
         $X_{id} = q - \beta * \text{abs}(m_{bestd} - x_{id}) * \log(1/u)$ ;
      else
         $X_{id} = q + \beta * \text{abs}(m_{bestd} - x_{id}) * \log(1/u)$ ;
      endif
    endfor
  endfor
endfor

```

It can be seen that the DCQPSO runs in attraction mode during the most of the iterations. Only when the diversity falls below the *dlow* do the particles fly in explosion mode. The explosion process is transitory, and once diversity is over the threshold, the population will return to attraction mode again.

5 Experiment Results and Discussion

We have tested the QPSO and DCQPSO on four widely known benchmark functions for testing the performance of different evolutionary optimization strategies. These functions are all minimization problems with minimum value zero. The four test functions are listed in Table 1. In all performance tests, the initial range of the population in all cases listed in Table 1 is asymmetry. Table 1 also lists V_{\max} for original PSO.

The fitness value is set as function value and the neighborhood of a particle is the whole population. The population size is set to be 20 for all cases. We test original PSO with inertia weight (called Standard PSO or SPSO), QPSO and DCQPSOs for performance Comparison. We had 50 trial runs for every instance and recorded mean best fitness. The maximum number of iterations is set as 1000, 1500 and 2000 generations corresponding to the dimensions 10, 20 and 30 for first three functions, respectively, The dimension of the last functions is 2 and the maximum number of iterations is 2000 for this function. In performance test of the SPSO, the inertia weight

Table 1. The table lists expression of benchmark function, the initial range of the population in the performance tests. The third column is the upbound of the velocity of the particle in the case of the SPSO.

Functions	Formulations	Initial Range	X_{max}	V_{max}
Rosenbrock function f_2	$f(x)_2 = \sum_{i=1}^n (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	(15, 30)	100	100
Rastrigrin function f_3	$f(x)_3 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	(2.56, 5.12)	10	10
Griewank function f_4	$f(x)_4 = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	(300, 600)	600	600
Shaffer's function f_5	$f(x)_7 = 0.5 + \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$	(30, 100)	100	100

Table 2. Average best fitness and standard deviation of all algorithms on Rosenbrock function

	Dimension	MAXITER	Mean Value	St. Dev.
SPSO	10	1000	94.1276	194.3648
	20	1500	204.337	293.4544
	30	2000	313.734	547.2635
QPSO	10	1000	59.4764	153.0842
	20	1500	110.664	149.5483
	30	2000	147.609	210.3262
DCQPSO1	10	1000	19.0109	29.5079
	20	1500	82.0134	84.4259
	30	2000	111.9926	164.3119
DCQPSO2	10	1000	34.6391	63.4889
	20	1500	102.1606	178.6908
	30	2000	128.0084	160.3456

Table 3. Average best fitness and standard deviation of all algorithms on Rastrigrin function

	Dimension	MAXITER	Mean Value	St. Dev.
SPSO	10	1000	5.5382	3.0477
	20	1500	23.1544	10.4739
	30	2000	47.4168	17.1595
QPSO	10	1000	5.2543	2.8952
	20	1500	16.2673	5.9771
	30	2000	31.4576	7.6882
DCQPSO1	10	1000	4.8308	2.4628
	20	1500	13.0424	4.8795
	30	2000	22.7104	5.5532
DCQPSO2	10	1000	4.6831	3.6387
	20	1500	15.3056	11.8478
	30	2000	24.2655	6.4856

w is decreases linearly from 0.9 to 0.4 as in [17]. In performance tests for QPSO and DCQPSOs, the Contraction-Expansion Coefficient β varies from 1.0 to 0.5 linearly when the algorithms are running. For DCQPSOs, this parameter control means that the population is in attraction mode. The threshold value of the diversity measure d_{low} is to be 0.005. Moreover, in DCQPSO1, the value of β_0 is set as 2.0.

The mean values and standard deviations of best fitness for 50 runs of each function are recorded in Table 2 to Table 5. The numerical results show that the DCQPSOs works better on the first three functions than QPSO and SPSO. DCQPSO1 has slightly better performance than DCQPSO2. On Shaffer’s function, the performances of DCQPSOs are not improved.

Table 4. Average best fitness and standard deviation of all algorithms on Greiwank function

	Dimension	MAIXTER	Mean Value	St. Dev.
		1000		
SPSO	10		0.09217	0.0833
	20	1500	0.03002	0.03255
	30	2000	0.01811	0.02477
QPSO	10	1000	0.08331	0.06805
	20	1500	0.02033	0.02257
	30	2000	0.01119	0.01462
DCQPSO1	10	1000	0.0781	0.0745
	20	1500	0.0189	0.0229
	30	2000	0.0090	0.0132
DCQPSO2	10	1000	0.0655	0.0464
	20	1500	0.0202	0.0204
	30	2000	0.0087	0.0122

Table 5. Average best fitness and standard deviation of all algorithms on Shaffer’s function

	Dimension	MAXITER	Mean Value	St. Dev.
SPSO	2	2000	2.78E-04	0.001284
QPSO	2	2000	0.001361	0.003405
DCQPSO1	2	2000	0.0012	0.0032
DCQPSO2	2	2000	0.0019	0.0039

6 Conclusion

In this paper, we proposed a method of controlling the diversity measure of the QPSO. The methodology of DCQPSO is setting a threshold value for the diversity to prevent the particles from clustering. Controlling of the diversity has been testified to be a good technique for enhancing the performance of the QPSO. However, for other benchmark functions, such as Sphere function and Ackley function, the results of this method is poor, since the optima of these functions is easy to find and the local search

ability is key to the performance of the algorithm. Annealing the diversity threshold d_{low} could lead to improvements, because it may be an advantage to decrease d_{low} near the end of the optimization.

References

1. P. J. Angeline, "Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and performance Differences," *Evolutionary Programming VII (1998)*, *Lecture Notes in Computer Science* 1447, pp. 601-610, Springer.
2. F. Van den Bergh, A. P. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer," 2002 IEEE International Conference on systems, Man and Cybernetics, 2002.
3. F. Van den Bergh, "An Analysis of Particle Swarm Optimizers," PhD Thesis. University of Pretoria, Nov 2001.
4. M. Clerc, "The Swarm and Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," *Proc. Congress on Evolutionary Computation* 1999, pp. 1951-1957.
5. M. Clerc and J. Kennedy, "The Particle Swarm: Explosion, Stability, and Convergence in a Multi-dimensional Complex Space", *IEEE Transaction on Evolutionary Computation*, no. 6, pp. 58-73, 2002.
6. A. P. Engelbrecht and A. Ismail, "Training product unit neural networks, " *Stability Control: Theory APPL.*, vol. 2, no. 1-2, pp.59-74
7. J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int'l Conference on Neural Networks, IV*. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942-1948.
8. J. Kennedy, "Stereotyping: Improving Particle Swarm Performance with cluster analysis," in *Proc. 2000 Congress on Evolutionary Computation*, pp. 1507-1512.
9. J. Kennedy, "Small worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance," *Proc. Congress on Evolutionary Computation* 1999, pp. 1931-1938.
10. P. N. Suganthan, "Particle Swarm Optimizer with Neighborhood Operator," *Proc. 1999 Congress on Evolutionary Computation*, pp. 1958-1962.
11. J. Sun et al, "Particle Swarm Optimization with Particles Having Quantum Behavior," *Proc. 2004 Congress on Evolutionary Computation*, pp. 325-331.
12. J. Sun et al, "A Global Search Strategy of Quantum-behaved Particle Swarm Optimization," *Proc. 2004 IEEE Conference on Cybernetics and Intelligent Systems*.
13. J. Sun et al, "Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level", *Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3049-3054.
14. R. K. Ursem: *Diversity-Guided Evolutionary Algorithms*, *Proceedings of The Parallel Problem Solving from Nature Conference* 2001.
15. J. Vesterstrom, J. Riget and T. Krink: *Division of Labor in Particle Swarm Optimization*. *IEEE 2002 Proceedings of the Congress on Evolutionary Computation*.
16. Y. Shi and R. Eberhart, "Empirical Study of Particle Swarm optimization," *Proc. of Congress on Evolutionary Computation*, 1999, 1945-1950.
17. Y. Shi, R. C. Eberhart, "A Modified Particle Swarm," *Proc. 1998 IEEE International Conference on Evolutionary Computation*, pp. 1945-1950.

Intelligent Agents as Cells of Immunological Memory

Krzysztof Cetnarowicz¹, Gabriel Rojek², and Rafał Pokrywka³

¹ Institute of Computer Science
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
cetnar@agh.edu.pl

² Department of Computer Science in Industry
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
rojek@agh.edu.pl

³ IBM SWG Laboratory*
ul. Armii Krajowej 18, 30-150 Kraków, Poland
rpokrywka@gmail.com

Abstract. Application of mechanisms of immune memory in the computer security domain allows to increase performance of certain class of security systems that are based on detection of attacks without *a priori* knowledge of attack's technique. Immune memory should enable the system to memorise once encountered attacks and prevent it together with its consequences in the future. The use of agent technologies gives new possibilities in the management of stored attack's patterns — patterns of obsolete attacks should be deleted but those of new and frequent should be maintained and generalised. In this paper ideas from agent technology and immune memory domain are introduced into computer security, tested and discussed.

1 Introduction

Work presented here is a part of research on Host Intrusion Detection System (HIDS) which is based on anomaly detection. This kind of security systems are usually placed in key nodes of network infrastructure. The system detects anomalies by comparing normal and current behaviour of monitored processes. This approach allows to detect an attack without *a priori* knowledge of attack's technique. To model the behaviour of a process the Markov model with variable order is used.

Here we would like to focus on mechanisms which will enable the system to record attacks and to use this recorded information to prevent the same or similar attacks in the future. In addition detection ought to be faster and the level of false alarms should decrease. All those merits should be achieved by the

* This paper is related to my knowledge and activities before I joined IBM.

realisation of mechanism similar to the natural immune memory. Agent technologies should allow to maintain stable in number population of recognisable patterns. The population should also dynamically adapt to all current threats by removal of agents responsible for detection of obsolete attacks and maintaining those responsible for detection of new and frequent ones. Agents use simple neural network to make decisions about particular threat. This should also enable them to generalise and detect similar attacks to those recorded. Similar attacks are those with only slightly changed technique.

2 Overview of the Natural Immune Memory

The human immune system is designed to protect an organism against pathogens. The most important elements of the system are lymphocytes which in great number circulate through the body and detect foreign cells. Basically there are two types of lymphocytes: T and B. They both cooperate in detection and elimination of pathogens. They also cooperate in creation of immune memory which we would like to model.

Immune memory is created during the primary immune response. This response is usually slow and has low intensity. The secondary immune response is initiated by *memory cells* and is very fast, intensive and often runs without any clinical symptoms of infection. Additionally the fact that it is triggered by infection with new, but similar to previously seen, pathogens means that immune memory is associative.

The primary response consists of several steps. First the lymphocyte of type B detects newly encountered pathogen. Then it receives confirmation of positive detection from lymphocyte T. This confirmation is called co-stimulation. Next it is activated and starts to mutate with high rate until it becomes specific to this one particular kind of pathogens. After the completion of this step the specific B-cell proliferate and differentiate into plasma cells and *memory cells*. The former secrete antibodies which participate in elimination of the pathogen, the later *memorise* the pathogen. Memory cells do not need any co-stimulation to become activated and to mount the secondary response, during which they rapidly proliferate and differentiate into plasma cells. This results in a fast increase of the number of antibodies and a very fast elimination of the threat. A detailed description of the entire immune system and can be found in [1], [4] and [5].

3 Mechanisms of Anomaly Detection

In this section a basic information about the HIDS used in our researches into immune memory is presented. The system is based on anomaly detection in behaviour of processes working on a given host. Behaviour is represented by sequences of system calls. Grounds for this choice can be found in [2] and [3].

Analysis of conditional entropy of the sequences suggests that it is reasonable to choose one of the statistical model which incorporates conditional probability to build a profile of these sequences. Here we have chosen the Markov chain with

variable order. This model provides a probability of the next system call having seen previous n . In this case n is a length of history and depends on the actual context — hence variable order. The profile is created during the training phase from the sequences of system calls representing normal behaviour according to the algorithm presented in [7]. An overview of Markov models can be found in [6].

Executing the next system call is a single step associated with the probability provided by the Markov model. During the monitoring phase this probability is mapped to some amount of penalty points. It is done through a penalty function $\xi : [0, 1] \rightarrow \mathbb{R}$. To be useful this function should give a large amount of penalty points for probability equal to 0 and small amount if the probability is close to 1. In our research the following function was used:

$$\xi(x) = \frac{a}{x+b} + c . \quad (1)$$

The a parameter controls how convex the function is, b and c parameters modifies the amount of points the function returns for a given probability. All three parameters must be carefully chosen. In each step the computed penalty points are added to a penalty account Ξ . To limit aggregation of history the penalty account is also multiplied by dumping factor $\zeta < 1$ (3).

$$\Xi_0 = 0 \quad (2)$$

$$\Xi_{i+1} = \zeta(\Xi_i + \xi_{i,i+1}) \quad (3)$$

where $\xi_{i,i+1}$ is the amount of the penalty points computed when going from i -th step to $(i+1)$ -th step. If the penalty account exceeds a certain threshold τ an alarm is fired. Figure 1 presents example plot of the penalty account and the threshold. From observations it can be concluded that each process has its own characteristic plot of behaviour what is a very important feature.

Subsequent values of the penalty account form a *signal* which is unique for particular process and is also unique for an anomaly. It is reasonable though to remember and analyse this signal and through this recognise intrusions. Anomaly detector described in this section is a first-line analyser. It fires an alarm whenever signal exceeds the threshold.

The main disadvantage of all anomaly detection methods is the possibility of false alarms. In addition detection is often too slow and attack or exploit could have been already executed. In this case any alarm has only informative value. Immune memory can speed up the moment of detection and thus possibly prevent the consequences of an attack. It can also lower the level of false alarms with time.

4 Intelligent Immunological Memory

The main idea of presented research is to realise memory cells as intelligent agents which are able to remember part of the process' signal (subsequent values of the penalty account). The memory agent is born in the moment of raising an

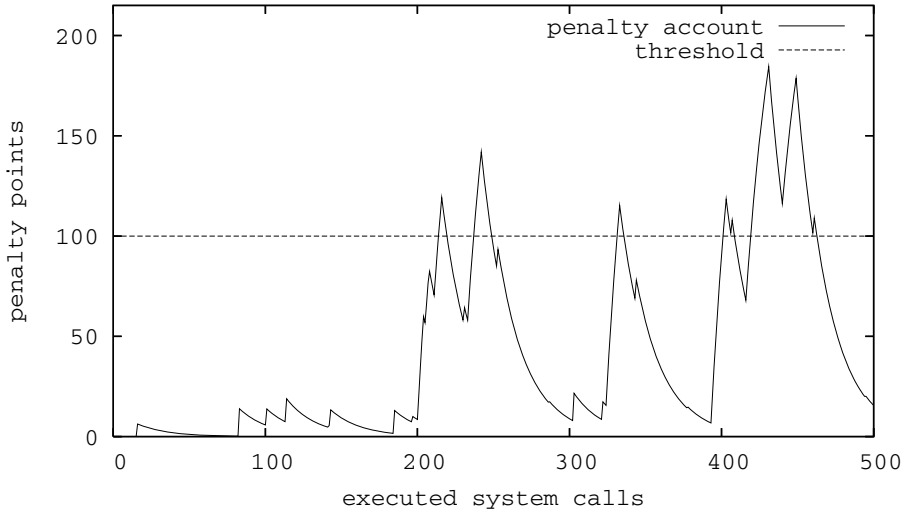


Fig. 1. Example plot of the penalty account and the threshold

alarm by the anomaly detector. Because of the possibility of false alarms this newly created agent needs a second signal to become fully functional. This signal comes from an administrator who must confirm that detected anomaly is really a threat. It is a kind of co-stimulation like in the natural immune system. This is the only place where human intervention is required. After the administrator's decision the agent is turned into one of the two types:

- *anomaly specific* — in case there was an anomaly (co-stimulation). The goal of this agent is to remember particular pattern of an attack and detect its future appearances.
- *anergic* — in case it was an false alarm (no co-stimulation). This agent's aim is to remember false alarm pattern and prevent future appearances of this false alarm.

Agents of both types use neural network to recognise patterns in the process' signal. Neural network also enables the agent to recognise similar patterns to that remembered. The anomaly specific agent has simply its neural network trained to remember a particular attack's pattern. Anomaly specific agents are able to initiate adequate response such as killing the process or tracking all execute calls. Anergic agents are specific to patterns of false alarms and if they manage to detect something no response should be initiated (there is an analogical mechanism in the natural immune system). Anergic agents are able to lower the level of false alarms with time. For IDS systems it is crucial to keep false alarm rate as low as possible. This is because usually the number of normal events in a system is significantly larger than the number of abnormal events. Normal events are therefore analysed more often and even if false alarm rate is low most raised

alarms turn out to be false. This problem is known as base rate fallacy and is directly implicated by Bayes' theorem. More details can be found in [8].

4.1 Life Energy

To maintain a stable in number population of agents and also to build a population which can dynamically adapt to current threats an idea of life energy is introduced. At the beginning an agent receives some fixed amount of life energy E_p . In each step the energy is decreased by 1. There is also determined a certain probability of death of the agent, which depends on the actual level of the agent's life energy. The probability is given by the following function:

$$p(x) = \begin{cases} -\frac{1}{E_s} + 1 & \text{for } x < E_s \\ 0 & \text{for } x \geq E_s \end{cases} \quad (4)$$

where E_s is an arbitrarily chosen value. When the agent happens to detect something its life energy increases also by arbitrarily chosen value E_a . This mechanism makes it possible to delete agents which detect obsolete patterns. It also keeps alive those agents responsible for detection of frequently occurring patterns.

4.2 Neural Network

The ADALINE (ADAPtive LINear Element) neuron is capable of recognising the most similar patterns to the remembered one what is a key feature for our purposes. Since the agent is (in this version) responsible only for detection of one particular pattern its neural network consists of one ADALINE neuron. The pattern is represented by a vector of m last values of the process signal. The neuron also has m inputs and m input weights. For now it is assumed that the length of the pattern is constant.

The activation of the neuron is given by the following formula:

$$y = \sum_{i=1}^m w_i x_i \quad (5)$$

where w_i is the i -th weight and x_i is the i -th input value. Inputs are often written simply as vector X and weights as vector W . From properties of ADALINE the more input vector X is similar to the weight vector W the greater the activation [9].

The neuron is learned through delta-rule which is an iterative process of adjusting the vector of weights according to the formula below:

$$W' = W + \eta \delta X \quad (6)$$

$$\delta = t - y \quad (7)$$

where η is the learning ratio, t is the target value that the neuron should give for this particular input vector X , y is the actual neuron's output for the input

vector X . The agent’s decision is positive when the current output is close enough to the target value, for example when it is $s = 99\%$ of the t . On the choice of s depends agent’s capabilities of generalisation. The s parameter defines also patterns similarity. In other words the agent groups similar patterns of similar anomalies.

5 Experimental Results

Datasets for tests were obtained from publicly available repository prepared by researchers from Computer Science Department, University of New Mexico [10].

The datasets consists of normal and anomalous traces of privileged processes. A trace is a sequence of system calls only.

To measure how fast an anomaly is detected a MTFA (Mean Time to First Alarm) metric is used. The MTFA is defined as:

$$\frac{\sum_{\rho \in S_{\text{an}}} \text{alarm}_1(\rho)}{|S_{\text{an}}|} \quad (8)$$

where S_{an} is a suite of anomalous traces, $|S_{\text{an}}|$ is the size of the suite and $\text{alarm}_1(\rho)$ is a number of system call when the trace ρ was first identified as anomalous. The rate of false alarms is measured simply as the ratio between the number of times normal trace is identified as anomalous and the number of times a trace was classified correctly.

In performed tests traces of sendmail were used. The normal database of this program consists of 346 traces and 1799764 system calls. Anomalous database consists of 25 traces 6755 system calls. There are traces of decode and sunsendmailcp exploits available and also an error condition — forwarding loop. There was 518229 normal system calls used in the training phase and 1281535 and all anomalous calls in the monitoring phase.

The testing phase begins when there is trained anomaly detector and empty agents population. The parameters for the anomaly detector were as follows: $\zeta = 0.95$, $\tau = 40$. For the penalty function it was assumed that in each step the maximal amount of penalty points was 15 and for probability 1 the function returned 0. Therefore the b and c parameters could be computed. The a parameter was set to 20. This parameter combination gave the best results and was chosen empirically. The population of anergic agents is created during runs through normal database and theirs E_p was chosen adequately to the size of the normal traces. Anomaly specific agents are created in the first and next runs through anomalous database. The E_s parameter was set to $\frac{E_p}{3}$ and $E_a = E_p$. Those parameters were also chosen empirically so that to achieve a stable agents population. The size of pattern was $m = 30$.

Tests have shown that specific memory agents detects anomalous traces faster than anomaly detector alone. It can be seen in a decrease in MTFA (Fig. 2). After the first run through normal database anergic agents have lowered false-alarm rate down to 0.0 (Fig. 3). Figure 5 presents a growth of the population of anergic agents with time. The number of agents doesn’t fall down since all

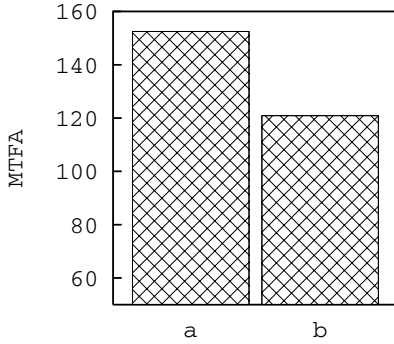


Fig. 2. Decrease in MTFA after the use of specific agents. The 'a' column refers to the result when no specific agents were used, 'b' when agents were used.

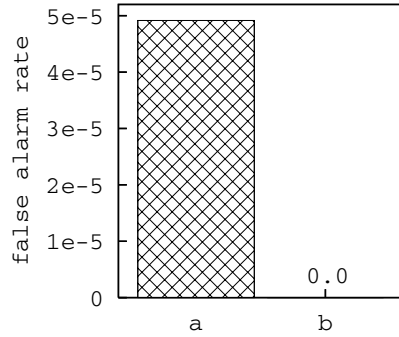


Fig. 3. Decrease in the level of false alarms when anergic agents are used. The 'a' column refers to the result without and 'b' with anergic agents.

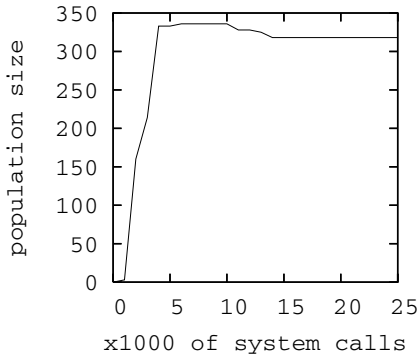


Fig. 4. The growth of population of specific agents

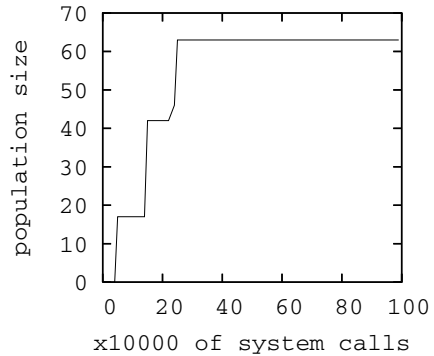


Fig. 5. The growth of population of anergic agents

patterns responsible for false alarms are present in the test suite. The growth of population of specific agents is presented on Fig. 4. At first all anomalous traces were included in the test suite what is reflected in the rapid growth and stabilisation of the population. Next, after two runs through the anomalous database, the traces of the decode exploit were removed. It can be seen that the population has shrunk adequately to the new situation.

6 Conclusions and Further Work

In performed tests it has been shown that memory agents are useful as an improvement of intrusion detection systems based on anomaly detection. They may help to prevent intrusions when a particular patch has yet to be developed.

Memory agents speed up the detection and lower false-alarm rate. Anergic agents are also useful when there is some change in program behaviour caused by, for example, library update. Other advantages include the reduction in amount of administrator's work and the adaptation of the agents population to the external conditions through mechanism of life-energy. However the disadvantage is that in real implementation the computer resources consumption is very high.

Tests presented here provide only a proof of concept and further work is needed. There are a few directions in which further research can go. First another potential benefits from using agent technologies must be mentioned. For example an ability to *immunise* other hosts in a LAN/WAN area. In this case a property of mobility must first be given to an agent. Additional benefits like better generalisation and prediction may be achieved by using specialized neural networks used in pattern recognition in financial analysis. This may result in further improve of MTFA. More investigation into identification by behaviour is also needed.

References

1. Hofmeyr, S.A.: An Interpretative Introduction to the Immune System. To Appear in the Design Principles for the Immune System and other Distributed Autonomous Systems, Oxford University Press (2000)
2. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.: A Sense of Self for Unix Processes. In IEEE Symposium on Security and Privacy (1996) pages 120–128
3. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion Detection Using Sequences of System Calls. Journal of Computer Security Vol. 6 (1998) pages 151–180
4. Wierzchoń, S.T.: Sztuczne systemy immunologiczne. Teoria i zastosowania. (in polish), EXIT, Warszawa (2001)
5. Kim, J. W.: Integrating Artificial Immune Algorithms for Intrusion Detection. PhD Thesis, Department of Computer Science, University College London (2002)
6. Bengio, Y.: Markovian Models for Sequential Data. Neural Computing Surveys 2 (1999) pages 129–162
7. Ron, D., Singer, Y., Tishby, N.: The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length. Machine Learning, 25(2–3) (1996) pages 117–149
8. Axelsson, S.: The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM Transactions on Information and System Security (TISSEC) 3 (2000) pages 186–205
9. Tadeusiewicz, R.: Sieci Neuronowe. (in polish), Wyd. 2, Akademicka Oficyna Wydawnicza RM, Warszawa (1993)
10. Data sets available on-line at <http://www.cs.unm.edu/~immsec/systemcalls.htm>. 2006

Negative Selection with Ranking Procedure in Tabu-Based Multi-criterion Evolutionary Algorithm for Task Assignment

Jerzy Balicki

Naval University of Gdynia, ul. Smidowicza 69,
81-103 Gdynia, Poland
J.Balicki@amw.gdynia.pl

Abstract. In this paper, an improved negative selection procedure to handle constraints in a multi-criterion evolutionary algorithm has been proposed. The problem that is of interest to us is the complex task assignment for a distributed computer system. Both a workload of a bottleneck computer and the cost of system are minimized; in contrast, a reliability of the system is maximized. Moreover, constraints related to memory limits and computer locations are imposed. Finally, an evolutionary algorithm with tabu search procedure and the improved negative selection is proposed to provide effective solutions.

1 Introduction

Evolutionary algorithms (EAs) have to exploit a supplementary procedure to incorporate constraints into fitness function in order to conduct the search precisely. An approach based on the penalty function is the most commonly used to satisfy constraints. Likewise, this technique is frequently used to handle constraints in multi-criteria evolutionary algorithms. However, it has some limitations, from which the most remarkable is the complicatedness to identify penalty coefficients.

The homomorphous mapping has been proposed as the constraint-handling procedure of EA to deal with parameter optimisation problems in order to avoid some impenetrability of the penalty function [10]. After that, a constrained-handling negative selection has been tested to optimisation problems with one criterion [3].

As a result, we propose an improved negative selection to handle constraints in multi-criterion optimisation problems. Both a workload of a bottleneck computer and the cost of machines are minimized; in contrast, a reliability of the system is maximized. Furthermore, constraints related to computer memory limits are imposed on the feasible task assignment. After all, an evolutionary algorithm with a tabu search procedure [12] and the negative selection is proposed to provide Pareto-optimal task assignments for the distributed systems. To avoid a limitation of the negative selection, we suggest introducing some distance measures from the state of an antibody to the state of the selected antigen, according to the constraints. In consequence, an improved negative selection with ranking procedure can be designed.

2 Negative Selection Algorithm

The immune system can be seen as a distributed adaptive system that is capable for learning, using memory, and associative retrieval of information in recognition [9]. Many local interactions provide, in consequence, fault tolerance, dynamism and adaptability [2]. The model of immune network and the negative selection algorithm are major outcomes on which most of the current works are based [8].

The negative selection algorithm (NSA) for detection of changes has been developed by Forrest et al. [6]. This algorithm is based on the discrimination principle that is used to know what a part of the immune system is and what it is not [5]. Detectors are randomly generated to reduce those detectors that are not capable of recognising themselves. Subsequently, detectors proficient to distinguish trespassers are kept. An adjusted detection is performed probabilistically by the NSA.

The negative selection can be used to manage constraints in an evolutionary algorithm by isolating the contemporary population in two groups [13]. Feasible solutions called “antigens” create the first cluster, and the second cluster of individuals consists of “antibodies” – infeasible solutions. For that reason, the NSA is applied to generate a set of detectors that verify the state of constraints.

We assume the fitness for antibodies is equal to zero. Then, a randomly chosen antigen G^- is compared to the selected antibodies. After that, the distance S between the antigen G^- and the antibody B^- is calculated due to the amount of similarity at the genotype level [1]:

$$S(G^-, B^-) = \sum_{m=1}^M s_m(G^-, B^-), \tag{1}$$

where

M – the length of the solution,

$$s_m = \begin{cases} 1 & \text{if } G_m^- \text{ is matching to } B_m^- \text{ at position } m, \\ 0 & \text{in the other case.} \end{cases} \quad m=1, \overline{M};$$

The measure of genotype similarity between the antigen and the antibody depends on their representation. This assessment of similarity for the binary representation (1) can be re-defined for the integer version, as follows:

$$S'(G^-, B^-) = \sum_{m=1}^M |G_m^- - S_m^-|. \tag{2}$$

The negative selection is a modified genetic algorithm in which infeasible solutions that are similar to feasible ones are preferred in the current population. Although, almost all the random choices are based on the uniform distribution, the pressure is directed to improve the fitness of appropriate infeasible solutions.

3 Ranking Procedure for Negative Selection

The situation that the fitness of the winner is increased by adding the magnitude of the similarity measure may pass over a non-feasible solution with the relatively small

value of this assessment. Nevertheless, some constraints may be satisfied by this alternative. What is more, if a constraint is exceeded and the others are performed, the value of a similarity measure may be low for some cases. That is, the first of two similar solutions, in genotype sense, may not satisfy this constraint and the second one may satisfy it.

To avoid this limitation of the NSA, it was suggested introducing some distance measures from the state of an antibody to the state of the selected antigen, according to the constraints. Equalities and inequalities that are of interest to us are, as follows:

$$g_k(x) \leq 0, \quad k = \overline{1, K}, \tag{3}$$

$$h_l(x) = 0, \quad l = \overline{1, L}. \tag{4}$$

The distance measures from the state of an antibody B^- to the state of the selected antigen G^- are defined, as below:

$$f_n(B^-, G^-) = \begin{cases} g_k(B^-) - g_k(G^-), & k = \overline{1, K}, n = k, \\ h_l(B^-), & l = \overline{1, L}, n = K + l, \end{cases} \quad n = \overline{1, N}, N = K + L. \tag{5}$$

The distance $f_n(B^-, G^-)$ is supposed to be minimized for all the constraint numbers n . If the antibody B^- is marked by the smaller assessment $f_n(B^-, G^-)$ to the selected antigen than the antibody C^- , then B^- ought to be preferred to C^- due to the improvement of the n th constraint. Moreover, if the antibody B^- is characterized by all the shorter distances to the selected antigen than the antibody C^- , then B^- should be preferred for all constraints. However, situations may occur when B^- is characterized by the shorter distances for some constraints and the antibody C^- is marked by the shorter distances for the others. In this case, it is difficult to select an antibody.

In this paper, we suggest introducing a ranking procedure to calculate fitness of antibodies and then to select the winner. In the previous works, a ranking idea for non-dominated individuals has been applied to avoid the prejudice of the interior Pareto alternatives [1, 3].

Now, we adjust this procedure to the negative selection algorithm (NSA*). Firstly, distances between the chosen antigen and some antibodies are calculated. Afterwards, the nondominated antibodies are determined according to their distances (5) to the antigen, and then, they get the rank equal to 1. Next, they are temporary eliminated from the population. Next, the new nondominated antibodies are found from the reduced population and they get the rank equal to 2. In this procedure, that level is increased and it is repeated until the subset of antibodies is exhausted. All the non-dominated antibodies have the same fitness because of the equivalent rank.

If B^- is the antibody with the rank $r(B^-)$ and $1 \leq r(B^-) \leq r_{\max}$, then the increment of the fitness function value is estimated, as below:

$$\Delta f(B^-) = r_{\max} - r(B^-) + 1. \tag{6}$$

Afterwards, the fitness of all the chosen antibodies are increased by adding their increments. The antibodies are returned to the current population and this process is repeated typically three times the number of antibodies. Each time, a randomly chosen antigen is compared to the same subset of antibodies. Next, the same procedure as for the NSA is carried out. Afterwards, a new population is constructed by reproduction, crossover and mutation without calculations of fitness. That process is repeated until a convergence of population emerges or until a maximal number of iterations is exceeded. At the end, the final population of the negative selection algorithm is returned to the external evolutionary algorithm.

4 Task Assignment in Distributed Computer System

To test the ability of NSA* for handling constraints, we consider a multi-criterion optimisation problem for task assignment in a distributed computer system [1].

Finding allocations of program modules may decrease the total time of a program execution by taking a benefit of the particular properties of some workstations or an advantage of the computer load. An adaptive evolutionary algorithm has been considered for solving multi-objective optimisation problems related to task assignment that minimize Z_{max} – a workload of a bottleneck computer and F_2 – the cost of machines [1]. The total numerical performance of workstations is another criterion for assessment the quality of a task assignment and it has been involved to multi-criterion problem formulated in [1]. Moreover, a reliability R of the system is an additional criterion that is important to assess the quality of a task assignment.

In the considered problem, both a workload of a bottleneck computer and the cost of machines are minimized; in contrast, a reliability of the system is maximized. In addition, constraints related to memory limits and computer locations are imposed on the feasible task assignment. A set of program modules $\{M_1, \dots, M_m, \dots, M_M\}$ communicated to each others is considered among the coherent computer network with computers located at the processing nodes from the set $W = \{w_1, \dots, w_i, \dots, w_I\}$. A set of program modules is mapped into the set of parallel performing tasks $\{T_1, \dots, T_v, \dots, T_V\}$. Some modern scheduling algorithms for tasks are proposed in [12].

Let the task T_v be executed on computers taken from the set of available computer sorts $\Pi = \{\pi_1, \dots, \pi_j, \dots, \pi_J\}$. The overhead performing time of the task T_v by the computer π_j is represented by an item t_{vj} . Let π_j be failed independently due to an exponential distribution with rate λ_j . We do not take into account of repair and recovery times for failed computer in assessing the logical correctness of an allocation. Instead, we shall allocate tasks to computers on which failures are least likely to occur during the execution of tasks. Computers and tasks can be allocated to nodes in purpose to maximize the reliability function R defined, as below [1]:

$$R(x) = \prod_{v=1}^V \prod_{i=1}^I \prod_{j=1}^J \exp(-\lambda_j t_{vj} x_{vi}^m x_{ij}^\pi), \tag{7}$$

where

$$x_{ij}^\pi = \begin{cases} 1 & \text{if } \pi_j \text{ is assigned to the } w_i, \\ 0 & \text{in the other case.} \end{cases}$$

$$x_{vi}^m = \begin{cases} 1 & \text{if task } T_v \text{ is assigned to } w_i, \\ 0 & \text{in the other case,} \end{cases}$$

$$x = [x_{11}^m, \dots, x_{1I}^m, \dots, x_{v1}^m, \dots, x_{vI}^m, x_{11}^\pi, \dots, x_{1J}^\pi, \dots, x_{ij}^\pi, \dots, x_{I1}^\pi, \dots, x_{Ij}^\pi, \dots, x_{IJ}^\pi]^T.$$

A computer with the heaviest task load is the bottleneck machine and its workload is a critical value that is supposed to be minimized [1]. The workload $Z_{\max}(x)$ of the bottleneck computer for the allocation x is provided by the subsequent formula:

$$Z_{\max}(x) = \max_{i \in 1, I} \left\{ \sum_{j=1}^J \sum_{v=1}^V t_{vj} x_{vi}^m x_{ij}^\pi + \sum_{v=1}^V \sum_{u=1}^V \sum_{i=1}^I \sum_{k=1}^I \tau_{vui k} x_{vi}^m x_{uk}^m \right\}, \quad (8)$$

where $\tau_{vui k}$ – the total communication time between the task T_v assigned to the i th node and the T_u assigned to the k th node.

Figure 1 shows the workload of the bottleneck computer in the distributed computer system for generated task assignments in a systematic way. The function Z_{\max} takes value from the period [40; 110] for 256 solutions. What is more, even a small change in task assignment related to the movement of a task to another computer or a substitution of computer sort can cause a relatively big alteration of its workload.

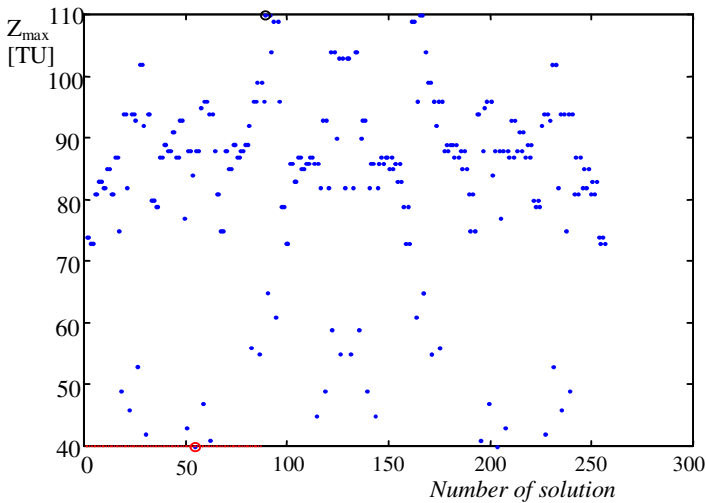


Fig. 1. Workload of the bottleneck computer for generated solutions

A computer should be equipped with necessary capacities of resources for a program execution. Let the following memories $z_1, \dots, z_r, \dots, z_R$ be available in an entire system and let d_{jr} be the capacity of memory z_r in the workstation π_j . We assume the task T_v reserves c_{vr} units of memory z_r , and holds it during a program execution. Both values c_{vr} and d_{jr} are nonnegative and limited. The memory limit in a machine cannot be exceeded in the i th node, what is written, as follows:

$$\sum_{v=1}^V c_{vr} x_{vi}^m \leq \sum_{j=1}^J d_{jr} x_{ij}^\pi, \quad i = \overline{1, I}, \quad r = \overline{1, R}. \tag{9}$$

The other measure of the task assignment is a cost of computers [1]:

$$F_2(x) = \sum_{i=1}^I \sum_{j=1}^J \kappa_j x_{ij}^\pi, \tag{10}$$

where κ_j corresponds to the cost of the computer π_j .

5 Adaptive Evolutionary Algorithm with NSA* and Tabu Mutation

The total computer cost is in conflict with the numerical performance of a distributed system, because the cost of a computer usually depends on the quality of its components. The faster computer or the higher reliability of it is the more expensive it is. Additionally, the workload of the bottleneck computer is in conflict with the cost of the system. Let (X, F, P) be the multi-criterion optimisation question for finding the representation of Pareto-optimal solutions [7]. It is established, as follows:

1) X - an admissible solution set

$$X = \{x \in \mathbf{B}^{I(V+J)} \mid \sum_{v=1}^V c_{vr} x_{vi}^m \leq \sum_{j=1}^J d_{jr} x_{ij}^\pi, \quad i = \overline{1, I}, \quad r = \overline{1, R};$$

$$\sum_{i=1}^I x_{vi}^m = 1, \quad v = \overline{1, V}; \quad \sum_{j=1}^J x_{ij}^\pi = 1, \quad i = \overline{1, I} \}$$

where $\mathbf{B} = \{0, 1\}$

2) F - a quality vector criterion

$$F : X \rightarrow \mathbf{R}^3 \tag{11}$$

where

\mathbf{R} – the set of real numbers,

$$F(x) = [-R(x), Z_{\max}(x), F_2(x)]^T \text{ for } x \in X,$$

$R(x), Z_{\max}(x), F_2(x)$ are calculated by (7), (8) and (10), respectively

3) P - the Pareto relation [14].

An overview of evolutionary algorithms for multi-objective optimisation problems is submitted in [3], [4]. An analysis of the task assignments has been carried out for two evolutionary algorithms. The first one was an adaptive evolutionary algorithm with tabu mutation AMEA+ [1]. Tabu search procedure was applied as an additional mutation operator to decrease the workload of the bottleneck computer. However, some numerical examples indicated that obtained task assignments have not satisfied constraints in many cases. Therefore, we suggest introducing a negative selection algorithm with ranking procedure to improve the quality of obtained solutions.

Better outcomes from the NSA* are transformed into improving of solution quality obtained by the adaptive multi-criterion evolutionary algorithm with tabu mutation AMEA*. This adaptive evolutionary algorithm with the NSA* gave better results than the AMEA+. After 200 generations, an average level of Pareto set obtaining was 1.5% for the AMEA*, 1.9% for the AMEA+ (Fig. 2).

Fifty test preliminary populations were prepared, and each algorithm started from these solutions. For integer constrained coding of chromosomes there were 12 decision variables in the test optimisation problem. The binary search space consisted of 1.0737×10^9 chromosomes and included 25 600 admissible solutions.

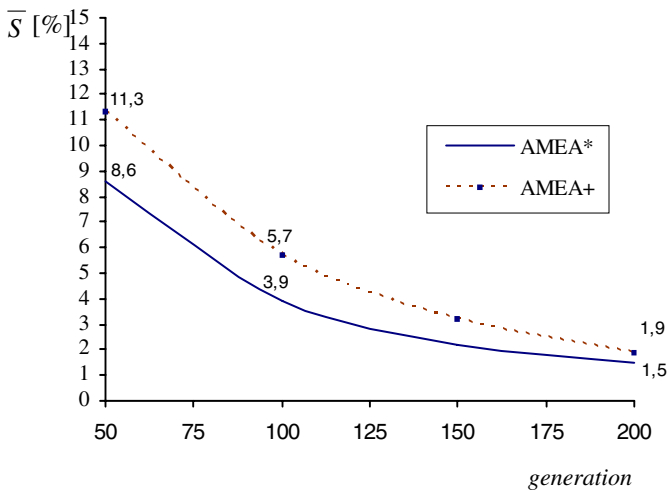


Fig. 2. Convergence for outcomes of the AMEA* and the AMEA+

6 Concluding Remarks

The adaptive evolutionary algorithm with the improved negative selection and tabu mutation can be applied for finding Pareto-optimal task allocations in a three-objective optimisation problem. In this problem, the workload of the bottleneck computer and the cost of computers are minimized. Moreover, the system reliability is maximised.

The negative selection algorithm can be used to handle constraints and improve a quality of the outcomes obtained by an evolutionary algorithm. Our future works

will concern on a development the AMEA* for finding Pareto-optimal solutions of the multi-objective optimisation problems.

References

1. Balicki, J., Immune Systems in Multi-criterion Evolutionary Algorithm for Task Assignments in Distributed Computer System. *Lectures Notes in Computer Science*, Vol. 3528 (2005) 51-56
2. Bernaschi, M., Castiglione, F., Succi, S.: A High Performance Simulator of the Immune System. *Future Generation Computer System*, Vol. 15 (1999) 333-342
3. Coello Coello, C. A., Van Veldhuizen, D. A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York (2002)
4. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester (2001)
5. Farmer, J.D.: A Rosetta Stone for Connectionism. *Physica D*, Vol. 22 (1990) 153-187
6. Forrest, S., Perelson, A.S.: Genetic Algorithms and the Immune System. *Lecture Notes in Computer Science* (1991) 320-325
7. Greco, S., Slowinski, R., Stefanowski J.: Mining Association Rules in Preference-ordered Data. *Lectures Notes in Computer Science*, Vol. 2366 (2005) 442-451
8. Jerne, N.K.: Idiotypic Networks and Other Preconceived Ideas. *Immunological Revue*, Vol. 79 (1984) 5-25
9. Kim, J. and Bentley, P. J.: Immune Memory in the Dynamic Clonal Selection Algorithm. *Proc. of the First Int. Conf. on Artificial Immune Systems*, Canterbury, (2002) 57-65
10. Koziel, S., Michalewicz, Z.: Evolutionary Algorithms, Homomorphous mapping, and Constrained Parameter Optimisation. *Evolutionary Computation*, Vol. 7 (1999) 19-44
11. Schaefer, R., Kołodziej, J.: Genetic Search Reinforced by the Population Hierarchy. In De Jong K. A., Poli R., Rowe J. E. (eds): *Foundation of Genetic Algorithms*, Morgan Kaufman Publisher (2003) 383-399.
12. Weglarz, J., Nabrzyski, J., Schopf, J.: *Grid Resource Management: State of the Art and Future Trends*. Kluwer Academic Publishers, Boston (2003)
13. Wierzchon, S. T.: Immune-based Recommender System. In O. Hryniewicz, J. Kacprzyk, J. Koronacki and S. T. Wierzchon (eds.) *Issues in Intelligent Systems. Paradigms*. Exit, Warsaw (2005) 341-356
14. Zitzler, E., Deb, K., and Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, Vol. 8, No. 2 (2000) 173-195

Multi-objective Optimization Using Co-evolutionary Multi-agent System with Host-Parasite Mechanism

Rafał Drezewski and Leszek Siwik

Department of Computer Science
AGH University of Science and Technology, Kraków, Poland
{drezew, siwik}@agh.edu.pl

Abstract. Co-evolutionary techniques for evolutionary algorithms are aimed at overcoming their limited adaptive capabilities and allow for the application of such algorithms to problems for which it is difficult or even impossible to formulate explicit fitness function. In this paper the idea of *co-evolutionary multi-agent system with host-parasite mechanism for multi-objective optimization* is introduced. In presented system the Pareto frontier is located by the population of agents as a result of co-evolutionary interactions between species. Also, results from runs of presented system against test functions are presented.

1 Introduction

Evolutionary algorithms (EAs) are techniques for finding suboptimal solutions of global optimization and adaptation problems, which are based on analogies to biological evolutionary processes. Evolutionary algorithms, however, often suffer from premature loss of population diversity. This results in premature convergence and may lead to locating local optimum instead of a global one. In the case of multi-modal problem landscapes EA without any special mechanisms will inevitably locate basin of attraction of single optimum. The loss of diversity also limits the adaptive capabilities of EAs in dynamic environments.

In *co-evolutionary algorithms* the fitness of each individual depends not only on the quality of solution to the given problem but also (or solely) on other individuals' fitness. This makes such techniques applicable in the cases where the fitness function formulation is difficult (or even impossible). Co-evolutionary techniques, are aimed at improving adaptive capabilities and introducing open-ended evolution into EAs by maintaining population diversity [8].

High quality approximation of *Pareto frontier* should fulfill at least three distinguishing features: first of all it of course should be "located" as close to the ideal Pareto frontier as possible what is very natural and common condition for both single- and multi- objective optimization, secondly it should include as many alternatives as possible and, at last, all proposed non-dominated alternatives should be evenly distributed over the whole ideal Pareto set.

In consequence, in the case of multi-objective optimization, premature loss of population diversity can result not only in lack of drifting to the ideal Pareto frontier but also in obtaining approximation of Pareto set that is focused around its selected

area(s) — what of course is very undesirable assuming that preference-based multi-objective optimization is not considered in this place.

Evolutionary multi-agent systems (EMAS) have proved their grate usefulness for solving a lot of different discrete, continuous, combinatorial and non-combinatorial multi-objective optimization problems [12, 11]. Co-evolutionary mechanisms are aimed at maintaining population diversity and improving adaptive capabilities of EMAS systems — especially in dynamic environments. This paper introduces the idea of *co-evolutionary multi-agent system with host-parasite mechanism for multi-objective optimization*. The process of locating Pareto frontier in such system emerges as a result of co-evolutionary interactions between species of agents. The results from runs of co-evolutionary multi-agent system for multi-objective optimization against commonly used test functions are also presented and the comparison to classical multi-objective evolutionary algorithms is made.

2 Evolutionary and Co-evolutionary Multi-objective Optimization

During most real-life decision processes a lot of different (often contradictory) factors have to be considered, and the decision maker has to deal with an ambiguous situation: the solutions which optimize one criterion may prove insufficiently good considering the others. From the mathematical point of view such multi-objective (or multi-criteria) problem can be formulated as follows [13].

Let the problem variables be represented by a real-valued vector:

$$x = [x_1, x_2, \dots, x_N]^T \in \mathbb{R}^N \quad (1)$$

where N gives number of the variables. Then a subset of \mathbb{R}^N of all possible (feasible) decision alternatives (options) can be defined by a system of:

- inequalities (constraints): $g_k(x) \geq 0$ and $k = 1, 2, \dots, K$,
- equalities (bounds): $h_l(x) = 0$, $l = 1, 2, \dots, L$

and denoted by \mathcal{D} . The alternatives are evaluated by a system of M functions (objectives) denoted here by vector $F = [f_1, f_2, \dots, f_M]^T$:

$$f_m : \mathbb{R}^N \rightarrow \mathbb{R}, \quad m = 1, 2, \dots, M \quad (2)$$

The key issue of optimality in the Pareto sense is the *weak domination relation*. Alternative x^a is dominated by x^b (which is often denoted by $x^b \geq x^a$) if and only if (assuming maximization of all objectives):

$$\forall m \ f_m(x^a) \leq f_m(x^b) \text{ and } \exists m \ f_m(x^a) < f_m(x^b) \quad (3)$$

A solution in the Pareto sense of the multi-objective optimization problem means determination of all non-dominated (in the sense of the defined above *weak domination relation*) alternatives from the set \mathcal{D} , which is sometimes called a *Pareto-optimal set*.

The Pareto-optimal set consists of globally optimal solutions, however there may also exist locally optimal solutions, which constitute locally non-dominated set (*local Pareto-optimal set*) [2]. The set $\mathcal{P}_{local} \subseteq \mathcal{D}$ is local Pareto-optimal set if [13]:

$$\forall x^a \in \mathcal{P}_{local} : \nexists x^b \in \mathcal{D} \text{ such that } x^b \geq x^a \wedge \|x^b - x^a\| < \varepsilon \wedge \|F(x^b) - F(x^a)\| < \delta \quad (4)$$

where $\|\cdot\|$ is a distance metric and $\varepsilon > 0, \delta > 0$.

The set $\mathcal{P} \subseteq D$ is global Pareto-optimal set if [13]:

$$\forall x^a \in \mathcal{P} : \nexists x^b \in D \text{ such that } x^b \geq x^a \quad (5)$$

These locally or globally non-dominated solutions create (in the criteria space) so-called local (\mathcal{PF}_{local}) or global (\mathcal{PF}) Pareto frontiers that can be defined as follows:

$$\mathcal{PF}_{local} = \{y = F(x) \in \mathbb{R}^M \mid x \in \mathcal{P}_{local}\} \quad (6a)$$

$$\mathcal{PF} = \{y = F(x) \in \mathbb{R}^M \mid x \in \mathcal{P}\} \quad (6b)$$

Multi-objective problems with one global and many local Pareto frontiers are called *multi-modal multi-objective problems* [2].

For the last 20 years a variety of evolutionary multi-criteria optimization techniques have been proposed. In the Deb's typology of evolutionary multi-objective algorithms (EMOAs) firstly the elitist and non-elitist ones are distinguished [3]. The main difference between these two groups of techniques consists in utilizing the so-called elite-preserving operators that give the best individuals (the elite of population) the opportunity to be directly carried over to the next generation regardless of the actual selection mechanism used. Deb's typology includes also so-called *constrained EMOAs*—i.e. algorithms and techniques that enable handling constraints connected with problem that is being solved.

Laumanns, Rudolph and Schwefel proposed co-evolutionary algorithm with predator-prey model and spatial graph-like structure for multi-objective optimization [6]. Deb introduced modified algorithm in which predators eliminated preys not only on the basis of one criteria but on the basis of the weighted sum of all criteria [3]. Li proposed other modifications to this algorithm [7]. The main difference was that not only predators were allowed to migrate within the graph but also preys could do it.

Co-evolution is the biological mechanism responsible for biodiversity and sympatric speciation. However it was not widely used as a mechanism of maintaining useful genetic diversity of population for evolutionary algorithms. It seems that co-evolution should introduce open-ended evolution, improve adaptive capabilities of EA (especially in dynamic environments) and allow speciation (the formation of species located within different areas of Pareto frontier or within local and global Pareto-frontiers in case of multi-modal multi-objective problems) but this is still an open issue and the subject of ongoing research.

3 Co-evolutionary Multi-agent System for Multi-objective Optimization

The main idea of *co-evolutionary multi-agent system (CoEMAS)* is the realization of species and sexes co-evolution in *multi-agent system (MAS)* [4]. CoEMAS model, as opposed to the basic *evolutionary multi-agent system (EMAS)* model [1], allows for the existence of several species and sexes which can interact with each other and co-evolve.

CoEMAS is especially suited for modeling different co-evolutionary interactions, such as resource competition, predator-prey and host-parasite co-evolution, sexual preferences, etc. Systems based on CoEMAS model can be applied, for example, to multi-modal function optimization and multi-objective optimization because such systems maintain population diversity and easily adapt to changing environment.

3.1 Co-evolutionary Multi-agent System with Host-Parasite Model

The essence of host-parasite approach consists in common evolutionary process (co-evolution) of two populations: population of *hosts* — representing problem solutions and population of *parasites* — representing tests that should be passed by *hosts*. *Hosts'* fitness value is proportional to the number of tests that each of them passed whereas *parasites'* fitness function value depends on number of *hosts* that do not pass test represented by given *parasite*. Of course each population can be characterized by its own: size, selection type, type of representation, genetic operators, probabilities of crossover and mutation etc. So, in another words, these are co-evolving but simultaneously independent populations.

Presented *co-evolutionary multi-agent system for multi-objective optimization with host-parasite mechanism* has been developed using *JagWorld* platform — a kind of Java-based infrastructure supplying basic mechanisms such as communication, parallelization etc. required during implementation systems according to both *EMAS* and *CoEMAS* model.

Realization of presented system required implementation of two kinds of agents: *host-agents* (representing solutions of problem that is being solved) and *parasite-agents* (representing "tests" for *host-agents* or rather for solutions represented by *host-agents*). The behavior of *host-agent* is similar to the behavior of "standard" agents characteristic for *EMAS-based* systems. So, *host-agent* "lives" in a place, it can move between places, and in every step it consumes resources needed for its life-activity. The fitness value is not directly assigned to the *host-agent* but it depends indirectly on interactions with population of *parasites* (*host-agents* representing worse solutions are more likely to be *infected* by *parasite-agents*). Each *parasite-agent*, similarly to the *host-agent*, consumes resources needed for living in the system in every step of simulation, but these agents do not receive resources from the environment, as it takes place in the case of *host-agents* but it takes resources from infected *host*.

The most distinguishing feature of *parasite-agent* is its possibility to infecting *host-agents*. In every step each *parasite-agent* that does not infect any *host-agent* tries to infect non-infected *host*. To infect a *host-agent* the *parasite-agent* performs specific test consisting in comparing objectives values represented by its genotype with objectives values of *host-agent* that is being infected. The probability of infection is higher or lower depending on performed test.

Both *host-agents* and *parasite-agents* can reproduce if they possess enough amount of resources. *Host's* reproduction consists in creating one descendant from two ready-for-reproduction individuals using crossover operator and then mutation operator is applied to created descendant. Parental individuals survive reproduction process but they lose some of their resources in aid of their offsprings. *Parasite's* reproduction consists

Table 1. Comparison of proposed CoEMAS approach with selected classical EMOA's according to the *Coverage of two sets* metrics

Coverage of two sets $\delta(A, B)$				
	SPEA	VEGA	NPGA	CoEMAS
SPEA	✓	0.08	0.00	0.04
VEGA	0.92	✓	0.30	0.32
NPGA	1.00	0.62	✓	0.40
CoEMAS	0.96	0.70	0.58	✓

in creating two descendants from one parental individual using mutation operator. Parental *parasite-agent* transfers half of its life-energy to each of its descendants and then dies.

At last, mentioned above test that is being performed by *parasite-agent* on *host-agent* before infection consists in comparing — in the sense of domination relation (see eq. (3)) — solutions represented by assaulting *parasite-agent* and *host-agents* that is being assaulted. The more solution represented by *host-agent* is dominated by *parasite-agent* the higher is the probability of infection.

3.2 Simulation Experiments — Preliminary Qualitative Results

After implementation some experiments have been performed, but because of space limitations only some qualitative conclusions (not quantitative results) will be here presented. Namely, proposed *co-evolutionary multi-agent system for multi-objective optimization with host-parasite mechanism* has been tested using, inter alia, *Binh* and slightly modified *Schaffer* test functions that are defined as follows:

$$F_1(\text{Binh}) = \begin{cases} f_1(x, y) = x^2 + y^2 \\ f_2(x, y) = (x - 5)^2 + (y - 5)^2 \\ \text{where } -5 \leq x, y \leq 10 \end{cases}$$

$$F_2(\text{Modified Schaffer}) = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x - 2)^2 \\ \text{where } -32 \leq x \leq 32 \end{cases}$$

Additionally, on the same *JagWorld* platform there have been implemented also some “classical” evolutionary algorithms for multi-objective optimization i.e. *Vector Evaluated Genetic Algorithm (VEGA)* [9, 10], *Niched-Pareto Genetic Algorithm (NPGA)* [5] and *Strength Pareto Evolutionary Algorithm (SPEA)* [13].

To compare proposed approach with implemented classical algorithms also some metrics have been used. Obtained values of these metrics are presented in Table 1, Table 2 and Table 3.

Assuming the following meaning of used below symbols: \mathcal{P} —Pareto set defined in eq. (5), $A, B \subseteq D$ —two sets of decision vectors, $\sigma \geq 0$ —appropriately chosen neighborhood parameter and $\|\cdot\|$ —the given distance metric, then the measures presented in these tables are defined as follows [13]:

Table 2. Comparison of proposed CoEMAS approach with selected classical EMOA's according to the *Coverage difference of two sets* metrics

Coverage difference of two sets $\xi(A, B)$				
	SPEA	VEGA	NPGA	CoEMAS
SPEA	✓	8	0	6
VEGA	116	✓	3	13
NPGA	154	42	✓	25
CoEMAS	197	27	7	✓

Table 3. Comparison of proposed CoEMAS approach with selected classical EMOA's according to another four metrics

	Size of dominated space (φ)	Average distance to the model Pareto set (M_1)	Distribution (M_2)	Spread (M_3)
SPEA	39521	0.8	0.21	10.2
VEGA	39405	2.3	0.11	10.3
NPGA	39368	3.2	0.18	10.1
CoEMAS	39324	3.7	0.15	9.9

– $\delta(A, B)$ —the coverage of two sets maps the ordered pair (A, B) to the interval $[0, 1]$ in the following way:

$$\delta(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \geq b\}|}{|B|} \tag{7}$$

– $\xi(A, B)$ —the coverage difference of two sets (φ denotes value of the *size of dominated space* measure):

$$\xi(A, B) = \varphi(A + B) - \varphi(B) \tag{8}$$

– M_1 —the average distance to the Pareto-optimal set \mathcal{P} :

$$M_1(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \min \{\|p - x\| \mid x \in \mathcal{P}\} \tag{9}$$

– M_2 —the distribution in combination with the number of non-dominated solutions found:

$$M_2(\mathcal{P}) = \frac{1}{|\mathcal{P} - 1|} \sum_{p \in \mathcal{P}} |\{r \in \mathcal{P} \mid \|p - r\| > \sigma\}| \tag{10}$$

– M_3 —the spread of non-dominated solutions over the set A :

$$M_3(\mathcal{P}) = \sqrt{\sum_{i=1}^N \max \{\|p_i - r_i\| \mid p, r \in \mathcal{P}\}} \tag{11}$$

Basing on defined above test functions and measures, some comparative studies of proposed co-evolutionary agent-based system and mentioned above very well known, and commonly used algorithms (i.e. *VEGA*, *NPGA* and *SPEA*) could be performed and conclusions from such experiments can be formulated as follows:

- Within the group of implemented algorithms *SPEA* has turned out to be definitely the best one;
- *NPGA* has turned out to be slightly worse than *SPEA* if the distance to the model Pareto frontier has been considered, and they have been very similar if distribution non-dominated individuals over the whole Pareto frontier has been considered;
- *VEGA*-based solutions have been almost as close to the model Pareto frontier as they have been in case of *SPEA* — however these solutions have been focused around some parts of Pareto set — what confirms the tendency of *VEGA* for preferring chosen objective(s);
- proposed *CoEMAS* system with *host-parasite* mechanism has turned out to be comparable to the *classical algorithms* according almost all considered metrics except for *Average distance to the model Pareto set* (see. Table 3);

It has to be mentioned here that preliminary experiments have been performed using very simple test functions and some potential advantages of proposed co-evolutionary system could not be here observed — but of course further experiments especially with very difficult multi-dimensional and dynamic testing problems will be conducted and proposed approach should turn out especially useful in case of multi-modal multi-objective problems such as Zitzler's t_4 test function [13].

4 Concluding Remarks

Evolutionary algorithms often suffer from premature loss of population diversity what limits their adaptive capabilities and possible application to hard problems like multi-modal and multi-objective optimization. To avoid such problems niching and co-evolutionary techniques for evolutionary algorithms are proposed and applied. However, co-evolutionary techniques are rather rarely used as mechanisms of maintaining useful population diversity.

The model of *co-evolutionary multi-agent system* allows co-evolution of several species and sexes. This results in maintaining population diversity and improves adaptive capabilities of systems based on *CoEMAS* model. In this paper the *co-evolutionary multi-agent system with host-parasite mechanism for multi-objective optimization* has been presented. The system was run against commonly used test problems and compared to classical *VEGA*, *SPEA*, and *NPGA* algorithms. Presented results show that *SPEA* is the best of all compared algorithms. Proposed *CoEMAS* with *host-parasite* mechanism was comparable to the other classical algorithms, except for *average distance to the model Pareto set* metric. This fact results from the tendency to maintain high population diversity what could be very useful in the case of hard dynamic and multi-modal multi-objective problems.

Future work will include more detailed comparison to other classical algorithms with the use of hard multi-dimensional, dynamic, and multi-modal multi-objective test

problems. Also the application of other co-evolutionary mechanisms like sexual selection and predator-prey are included in future plans.

References

1. K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki. The application of evolution process in multi-agent world to the prediction system. In M. Tokoro, editor, *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS 1996)*, Menlo Park, CA, 1996. AAAI Press.
2. K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.
3. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
4. R. Dreżewski. A model of co-evolution in multi-agent system. In V. Mařík, J. Müller, and M. Pěchouček, editors, *Multi-Agent Systems and Applications III*, volume 2691 of *LNCS*, pages 314–323, Berlin, Heidelberg, 2003. Springer-Verlag.
5. J. Horn, N. Nafploitis, and D. Goldberg. A niched pareto genetic algorithm for multi-objective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, 1994.
6. M. Laumanns, G. Rudolph, and H.-P. Schwefel. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature — PPSN V*, volume 1498 of *LNCS*. Springer-Verlag, 1998.
7. X. Li. A real-coded predator-prey genetic algorithm for multiobjective optimization. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization, Second International Conference (EMO 2003), Proceedings*, volume 2632 of *LNCS*, pages 207–221. Springer-Verlag, 2003.
8. J. Paredis. Coevolutionary algorithms. In T. Bäck, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation, 1st supplement*. IOP Publishing and Oxford University Press, 1998.
9. J. D. Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University, 1984.
10. J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, 1985.
11. L. Siwik and M. Kisiel-Dorohinicki. Balancing of production lines: evolutionary, agent-based approach. In *Proceedings of Conference on Management and Control of Production and Logistics*, pages 319–324, 2004.
12. K. Socha and M. Kisiel-Dorohinicki. Agent-based evolutionary multiobjective optimization. In *Proceedings of the Congress on Evolutionary Computation*, pages 109–114, 2002.
13. E. Zitzler. *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 1999.

Development of Multi Agent Resource Conversion Processes Model and Simulation System

Konstantin A. Aksyonov¹, Elena F. Smoliy¹,
Natalia V. Goncharova¹, Alexey A. Khrenov¹, and Anastasia A. Baronikhina²

¹The Ural State Technical University, box 305 "To Aksyonov",
620078 Ekaterinburg, Russia
wiper99@mail.ru

²The Ural State Technical University, Ekaterinburg, Russia
nastya_baro@mail.ru

Abstract. The mathematical model of multi agent resource conversion processes (RCP) is developed by the means of discrete-event simulation systems and expert systems. Within the framework of mathematical model RCP are defined: production system of the RCP structure, that taking into account conflicts origin. The discrete-event simulation system "BPsim" is developed on the basis of the multi agent RCP mathematical model. The "BPsim" system is inculcated on the firms in Ural region (Russia).

1 Introduction

One of the perspective directions of simulation and modeling (SM) tool development is the problem-orientation, which allows to reduce the requirements that presented to a knowledge level of end users in the programming area essentially. In the given work the results of creation and application of the resources conversion processes theory and SM system "BPsim", created on the basis of that theory, are described.

2 Mathematical Model of the Multi Agent Resources Conversion Process (RCP)

In this research, we shall understand the resources conversion process as the process of an input conversion (resources necessary for process execution) in an output (products - outcomes of process execution).

The tasks of problem area RCP are: new RCP designing and existing RCP perfecting, the forecast of resources and conversion device state, the estimation of the process time and cost characteristics, the resources costs estimate and mechanisms usage time.

To the basic disadvantages of the visual problem-oriented SM tools such as Arena, ARIS, ReThink (G2), with reference to the RCP, it is possible to attribute the complexity of the RCP description and carrying out of computing experiments, weak modeling means of the conflicts arising with the general resources and tools, absence of Russian language support. In this paper we examine mathematical model of the

RCP and SM system “BPsim”, which is substantially free from the above mentioned disadvantages.

Creation of the RCP mathematical apparatus is based on the widespread mathematical schemas of dynamic processes description (Petri networks [1], queuing system [1, 2], models of system dynamics [1, 2, 3]). However, with the help of the specified models it is difficult enough to present all the features of the RCP.

The main objects of discrete Multi Agent RCP are (Fig. 1): operations (*Op*), resources (*Res*), control commands (*U*), conversion devices (*Mech*), processes (*PR*), sources (*Sender*) and resource receivers (*Receiver*), crossroads (*Junction*), parameters (*P*), agents (*Agent*). The process parameters are set by the object characteristics function. The relations between resources and conversion device are set by the link object (*Relation*). The agents existence resumes availability of the situations (*Situation*) and decisions (action plan) (*Decision*).

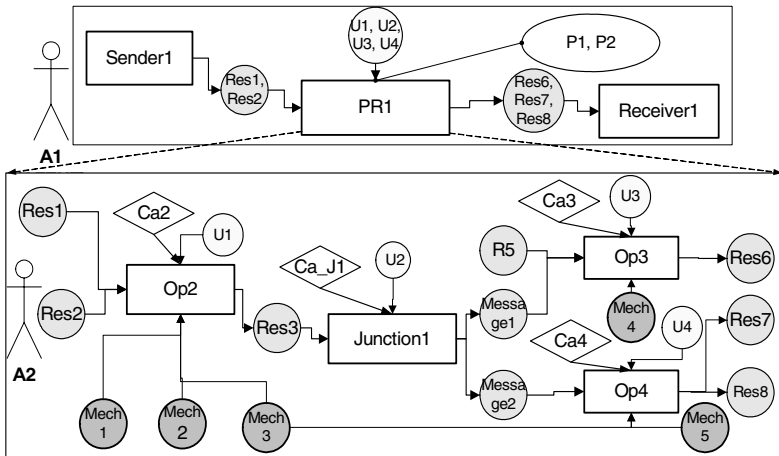


Fig. 1. Hierarchical Multi Agent RCP. PR1 process works out in constituents Op2, Op3, Op4, Junction1 and corresponding resources and tools; agents A1 and A2 control different RCP levels

Agents operate the RCP objects. For every agent there is a model of the person taking a decision. An agent executes the following actions: 1) analyzes the environment (current situation); 2) diagnosis the situation, apply to the knowledgebase. In the case of determine any corresponding (exceptional) situation an agent try to find a decision (action scenario) in the knowledgebase or work out it himself; 3) works out (take) a decision; 4) determines (remolds) the goals; 5) controls goals achievement; 6) delegate the goals to his own or somebody else's RCP objects, and also to another agents. 7) exchanges messages.

The RCP elements take part in message exchange and realize their converter functions on the base of their behavioral models (state graphs) following the incoming messages. The frames are selected as an agent's knowledge representation language.

The possibility of the use of typical process description models for the creation of the mathematical model of the RCP is examined; the typical process description

models are: the augmented Petri networks; the queuing systems. It's shown that given models don't allow making an adequate representation of the RCP. There are some disadvantages revealed for the Petri networks: absence of timing; absence of the concurrent activities conflicts; the lack of the division of mark types (resources types); models of real processes described in the terms of Petri networks are bulky and badly readable. Owing to the fact that the change of the N_E – schemas has only two positions, there is no ability of process interrupt modeling. The conceptual apparatus of the Q-schemas are not corresponding with the problem area of the RCP, Q-schemas are oriented on the modeling instruments activity, and in the RCP there are modeling of the consecution and parameters of the conversion processes.

The apparatus production rules are used for building a kernel of simulation system [4, 5]. The structure of the production rules of the RCP system is defined as:

$$PS = \langle Rps, Bps, Ips \rangle . \tag{1}$$

In equation (1) $Rps = \{RES(t)\} \cup \{MECH(t)\}$ is the current state of resources and conversion devices (operative storage); Bps is a set of resources transformation rules (knowledge base); Ips is an inference engine.

The operation algorithm of the inference engine consists of the following main stages: 1) definition of a current instant $SysTime = \min T_j, j \in \{RULE\}$ (standard algorithm of the discrete-event simulation (DES)); 2) agent's actions processing (current situation diagnosis, executive instruction working-out); 3) queueing of transformation rules; 4) execution of transformation rules and operative storage state transition. Imitator applies to the expert system module for the current situation diagnosis and executive instruction working-out.

3 Multi Agent Simulation System “BPsim”

The problem-oriented SM system is developed on the basis of surveyed model. SM package of the RCP “BPsim” is worked out on the basis of the following means:

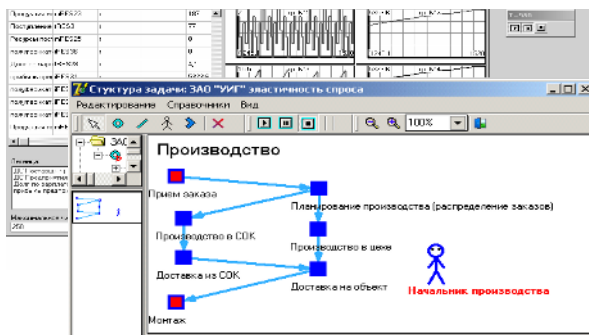


Fig. 2. A fragment of the industrial process model which is worked out for the “Urals Industrial Group” enterprise and made with the help of the simulation system “BPsim”. For every graph node in the form view there is an operation or RCP, for every connection between the nodes there is a resources flow, for every agent there is a model of the director of operations.

Borland Delphi 7 and database control system MS SQL Server. The SM system “BPsim” provides execution of the following functions: the creation of dynamic model RCP [6]; the simulation (Fig. 2); the analysis of the simulation experiment results; obtaining reports on the results; export of the experimental results in MS Excel and MS Project formats.

4 The “BPsim” System Application

Some tasks of the evaporator machine-building production process which count 420 technological operations was solved for the large chemical engineering Urals plant “URALKHIMMASH”: planning of the operations schedule; estimation of equipment loading; estimation of man power loading. The dynamic model of activity with the “BPsim” was designed for “The Ural industrial group”. The use of the designed system allows defining the optimum schedule of windows delivery and mounting links operation; the optimum quantity of mounting links depending on seasonal demand (volumes of orders); and also it allows reducing the periods of execution of orders from 14 days to 11, that helps to increase profits of the firm by 21 %.

5 Summary

The mathematical model of the multi agent RCP is developed on the basis of the means of dynamic expert systems. Within the framework of mathematical model multi agent RCP are defined: production system of the RCP structure.

The package of multi agent simulation modeling “BPsim” is developed and inculcated on the firms in Ural region and in the Ural State Technical University.

The multi-agent method application to the dynamic model of the RCP helps to increase its intellectuality owing to expert and simulation fusion.

References

1. Avramchuk E.F., Vavilov A.A., Emelianov S.V. etc.: Technology of system simulation, M.: machine construction industry; Berlin: Techniques, 520 p, under edition (1988)
2. Pritsker, A. A. B.: Introduction to simulation and SLAM II, System Publishing Corporation., West Lafayette (1984)
3. Jay Forrester: Industrial Dynamics. Cambridge, MA: MIT Press (1961)
4. Peter Jackson.: Introduction to Expert Systems, West Group, Rochester, NY, Addison-Wesley (1998)
5. A. Newell: Production systems: models of control structures // Visual information processing, New York: Academic Press (1973) 463-526
6. K. Aksyonov, B. Klebanov, A.Khrenov; Computer-aided design system of simulation business process model, Proceedings of the 4th IMACS Symposium on Mathematical Modeling, ARGESIM Report no. 24. – Austria, Viena University of Technology (2003) 1414-1420.

Designing Floor-Layouts with the Assistance of Curious Agents

Ewa Grabska¹, Katarzyna Grzesiak-Kopec², and Grażyna Ślusarczyk¹

¹ Jagiellonian University, Institute of Computer Science, Nawojki 11,
30-072 Cracow, Poland

uigrabsk@cyf-kr.edu.pl, grazyna@ii.uj.edu.pl

² Jagiellonian University, The Faculty of Physics, Astronomy and Applied Computer
Science, Department for Information Technology,
Reymonta 4, 30-059 Cracow, Poland
katarzyna.grzesiak-kopec@uj.edu.pl

Abstract. The paper deals with visual computational design in which emergence is a key to creativity. The presented framework for conceptual design uses shape grammars and curious agent assistants. The intelligent agents perceive the changing environment and emergent phenomena that occur in it. Interacting with each other and the designer they look for the most original and plausible solutions to a given design task. The approach is illustrated by the example of a designing floor-layouts.

1 Introduction

This paper presents a new framework for visual kind of creative conceptual design with shape grammars and curious agent assistants. It takes advantage of visual shape grammar computations, emergent phenomena which enhance creativity and curiosity of intelligent agents searching for novel solutions. The approach deals with various external conditions while the design process takes place. The interaction with the environment can happen both through the autonomous designer's decisions and through the approved curious agents prompts. The framework is illustrated by the example of a designing floor-layouts.

2 Dynamic Creative Design and Shape Grammars

One of the most important phenomena which enhances creativeness is emergence. In design, these features which are not explicitly represented but emerge from a design structure are called emergent (Fig. 1). The ability to recognise properties which are not originally intended in design is an important aspect of human visual perception and plays a key role in the interaction between the designer and his/her drawing or sketch. In most computational systems emergence is limited to generation and observation of emergent features. Using shape grammars as generative systems allows us not only to generate and recognise emergent elements but fed them back into computation as well [4]. The formal

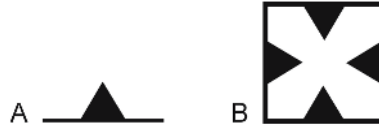


Fig. 1. (A) Single object. (B) Configuration of four copies of the given object which implies an emergent cross.

model [2] for dynamic design using shape grammars has been verified by a dedicated application *DesignAnalyzer2D* [3]. The classified emergent elements [2] are detected on-line by the system and new design rules containing them (emergent rules) are defined. We have programmed shape grammars with the use of a control diagram which specifies the order of applying grammar rules.

3 Curious Design Agents Architecture

In our approach agents are to be the designer's assistants in creative design. By an *agent* we understand a computer system *situated* in an *environment* and capable of autonomous actions in this environment in order to satisfy its design task [6]. Its inside structure consists of processes connected with perception, evaluation, planning, and action which affects the environment. By creative designs we understand here designs which are in a way new (unexpected, original) and useful (valuable, valid) [1]. In order to perform the task, each curious agent has to be *intelligent*: perceive its environment and respond to changes that occur in it, exhibit goal-directed behaviour and interact with other agents [7].

A curious agent observes its environment, gathers perceptual inputs and changes its internal state appropriately. On the one hand, the goal of our agent is to determine the novelty of a situation which means to validate the originality of the current environmental state with respect to the history. On the other hand, on the basis of the agent's interest in the present situation it has to identify unexpected consequences of actions. Thus, the agent's interest depends on the gained knowledge, while its curiosity can be understood as a motivation to do the further research. To support curiosity, the minimal agent architecture must include some form of memory. The mentioned requirements suggest using two different types of memory: a short-term memory to store recent experiences and a long-term one to store agent's generalised experiences from the past [5]. Both novelty and surprise of the situation can be computed by comparing the current state of an agent with its previous experiences. Information about the environment state and history, and the interest degree are taken into account in a decision making process. New goals of an agent are specified and the chosen goal is mapped into an action. In the proposed solution the short-term memory is implemented as a store of variables. The long-term memory has a form of a neural network called Self-Organising Map (SOM). It should be stressed that different implementations of a short-term and a long-term memory influence the learning ability of the agent and can lead to different curious behaviour.

4 Curious Agent Assistant

We expect that the curious agent assistants extend our possibilities to obtain interesting and valid design solutions. Two curious agent assistants are presented to support the designer in searching for original floor-layout designs by promoting the most creative solutions to a given design task.

Our environment consists of a set of design rules with a control diagram specifying the order of their application and the artefact to be designed in a given phase of generation. The environment is inspected by two curious agent assistants: *inquisitive (i-agent)* and *sage (s-agent)*. The i-agent is responsible for introducing new emergent rules into a shape grammar and the s-agent is in charge of a generation of possible designs. In order to improve the learning process of the agents and to incorporate the designer knowledge and preference more efficiently, a list of the best solutions, proposed by agents, is presented to the designer and he/she chooses one of them. Thus, the agents future decisions depend not only on their mutual interactions but on the interaction with the designer as well.

Let us consider the example of designing a floor plan. The designer has to determine two initial requirements: an overall shape of a floor plan (Fig. 2A) and a number of rooms. Assume that the required number of rooms is equal to 5. Applying design rules shown in Fig. 2, in a sequence 1, 3, 2, 3 the designer obtains an initial floor-layout with 5 rooms (Fig. 2B).

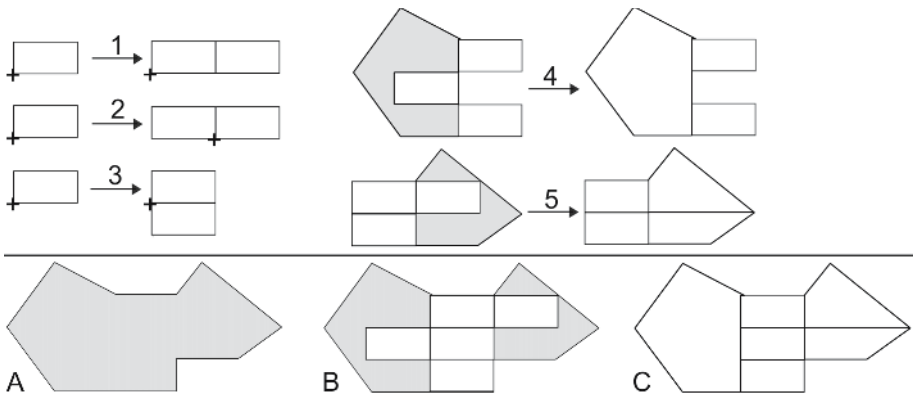


Fig. 2. (1-3) Regular grammar rules; (4,5) Emergent rules; (A) Given floor plan; (B) Early floor-layout with 5 rooms generated; (C) The designed floor-layout

The i-agent looks for new interesting design rules based on emergent elements which have come on the scene in the current step of generation. It localises emergent holes [2] (empty spaces), determines the design elements which constitute them and generates a set of possible rules to replace them. Then, basing on its experience and knowledge, it proposes the most appropriate design rules to the s-agent (Fig. 2 rule numbers 4 and 5).

The s-agent evaluates the rules proposed by the i-agent by applying them to the current generation and verifying whether the obtained solution is still valid. For example, it checks whether the generated floor-layout does not include overlapping areas or the number of rooms is different from the required one. Knowing the current generation step, the s-agent takes into account also rules which are already in the grammar, and evaluates them in the same manner as emergent ones. Having compared both emergent and regular solutions it recommends the best one, which satisfies all conditions, to the designer as the next step in the present generation. It also points out the shapes in the generation to which the rule should be applied. Fig. 2C shows one of the possible final solutions after applying the following rules: 1, 3, 2, 3, 4, 5.

Now, let us consider alternative flows of events. Since not all new rules proposed by the i-agent are acceptable to the s-agent, there may be more than one iteration in the negotiations between two assistants. If all attempts at mediation fail, either the best from the obtained solutions or none of them is chosen.

5 Conclusions

In this paper we presented a new approach to conceptual design. In computational design the emphasis on the dynamic design context is crucial to artificial creativity. Visual computations are situated in a context indirectly through the designer working with the application or more explicitly through curious agents. The intelligent agents perceive the changing environment and emergent phenomena that occur in it. Interacting with each other and the designer they look for the most original and plausible solutions to a given design task.

References

1. Grabska, E., Ślusarczyk, G. and Grześ, P.: Dynamic design with the use of intelligent agents. Proceedings of the 4th International Conference on Computer Recognition Systems CORES'05. Springer (2005) 827–834
2. Grzesiak-Kopec, K.: Emergent elements in periodic designs: An attempt at formalization. Design Computing and Cognition '04. Kluwer Academic Publishers (2004)
3. Grzesiak-Kopec, K.: Visual designing of graphical models with the use of emergent elements. PhD thesis (in polish). Faculty of Electrical Engineering, Automatics, Computer Science and Electronics. AGH University of Science and Technology (2005)
4. Knight, T.: Interaction in visual design computing. Visual and Spatial Reasoning in Design III. The invited paper (2004)
5. Saunders, R.: Curious design agents and artificial creativity. PhD thesis. Faculty of Architecture. The University of Sydney (2002)
6. Wooldridge, M. J. and Jennings, N.R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review. **Vol. 10(2)** (1995)
7. Wooldridge, M. J.: Intelligent agents. Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press. Cambridge. MA (1999)

Supporting Software Agents by the Graph Transformation Systems

Leszek Kotulski

Institute of Computer Science Jagiellonian University
kotulski@ii.uj.edu.pl

Abstract. Agent systems demand from graph transformation systems not only an effectiveness of parsing but also supporting of the online derivation control (in order to visualize or control of their development). The way of the effective realization Derivation Control Environment for the distributed systems is presented. Finally, the computational effectiveness of the parsing, membership problems of the full graph transformation system is discussed.

1 Introduction

Multiagent systems consist of an agents environment and a number of software agents that are situated in that environment. Agents act autonomously, driven by their goals and plans, thereby sensing and reacting to their environment and cooperating with other agents. This environment represents not only hardware architecture but also the some logical resources (such as data bases, message subsystem, etc.) necessary to preform the agents task [9]. An individual agent can be executed in the computing unit if the proper local environment is present in this unit; otherwise either this agent should be moved to another computing unit or this unit environment has been enriched. Let us notice, that computing units (connected by communication media), resources associated with each environment and agent system can be represented, every separately, as a graph structures. The connection of the above structures is not only difficult to perform¹ but is also extremely difficult to visualize and manage.

The graph transformation mechanism seems to be useful in the support both the analysis some system properties and the online control of the distributed agent system description (DASD) modification. The first task was a matter of intensive and and finished with the success research [2]. These techniques are especially useful when we have a graph, which describes some problem, and we would like to answer if it belongs to the class of objects described by a given graph grammar. It is necessary to mark, that only few grammars can solve the membership problem with the polynomial computational complexity (as for example [6]). The graph transformations become also an important formal specification technique [3].

¹ We cannot simple join these graphs and trees because graph isomorphism in general case is an NP-complete problem.

The online control of the DASD modification needs the introduction of the derivation control mechanism, which is able online react on the events appearing in the distributed network. Such a proposition is presented in section two. Next, the usefulness of this proposition is illustrated by two examples: the coordination of changes in the agents environment and the optimization of the agent system effectiveness. For large systems it would be desirable if this control be held in the parallel way. Thus, at the end the parallel derivation and time complexity of the proposed solution is considered.

2 Derivation Control Environment

It is already proved that the graph transformations can be useful for problems describable by graphs, that properties should be checked. Problems appear when we need control dynamic changes occurring in the distributed environment as the reaction on an individual activity of its components. Unfortunately, individual components activity and their active cooperation are the key concept of the agent systems [1, 10].

We assume, that the minimal requirements to the graph transformation system useful for agent system seems to be following: react on-line to both user and software requests, solve a problem of the derivation control in a parallel way, be able to parse graphs with the polynomial computational complexity for grammars with sufficient descriptive power to solve problems at hand.

The last requirement can be fulfilled with an ETPL(k) graph grammar, used in such different areas as pattern recognition [6] and on-line allocation control [5]. The $O(n^2)$ computational complexity of parsing has been proved for this grammar [4]. To fulfill the rest of the above requests there is a need to design a special environment making possible an on-line parallel derivation control, which is called the **derivation control environment (DCE)**. The basic term in this environment is the term **request**.

The **requests** are serviced by a DCE, that can be both monolithic and distributed one. A distributed DCE is a collection of monolithic DCEs, which communicate via system request. Formally DCE is introduced in [7]. Intuitively (see Fig. 1), the derivation control diagram can be interpreted as a graph connecting **Control Points** (the dotted circles) inside which sequentially are evaluated the synchronizing function (if exists) and the selector choosing one of the transitions from the active **CP** to another one (drawn as an edge).

During such transition, both the production $P_{k,i}$ is applied and the semantics action $SF_{k,i}$ is executed. The semantics action **SF** (associated with the transition): enriches the order queue (requesting some

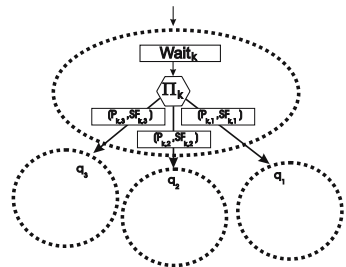


Fig. 1. Part of DCE

actions of the operating system), removes the **request** (that is serviced) from the request queue, evaluates parameters of the right-hand graph of the production P. The production P is applied to the current graph G creating a new graph G', in a way that is defined by transformation rules of the graph grammar associated with this derivation control diagram.

3 Derivation Control Environment and Agent Systems

Let us consider an example of usefulness of the graph transformation mechanism in the case of agent system migration problem. We assume that an agent can migrate in a distributed computing environment in order to find the least overloaded computing node. Unfortunately, the anomaly takes place when a new computing node is starting: all agents want to migrate to this node what immediately overload it. We assume that, DCE maintains the current allocation graph, and agent migration is associated with performing some productions updating this graph. The overloading problem can be solved if before migration each agent asks DCE for a permission [8]. This permission will be given only for a few agents. Moreover, such DCE can also be treated as an agent introducing the migration policy on the top level, by introducing some optimization algorithm. The ordered structure of the graph makes possible to complete such an optimization with the polynomial computational complexity of ($O(n^2)$) [7].

The second example is based on the problem pointed in [9]. We assume that a multiagent application consists of set of virtual machines (creating an local environment) and number of agents. The virtual machine enables for example the access to the local data base system by an agent. When an agent needs move to another destination it is limited only to this nodes which offers the same (or greater) virtual machine properties. Similarly as in the preceding example we assume that DCE maintains the allocation graph; its structure is however more complicated, there are represented there layers: hardware, virtual machines and agents. The role of DCE is in this case more complicated, and it can not be solved by the execution of two productions (removing agents node and putting it in another place). Application based on DCE have to check whether new virtual machine offers at least the same properties as the old one (it can be made using selectors). Otherwise, we have to enrich this environment before moving the agent to it. This task needs direct support of the graph grammar properties; as the old virtual machine is represented as a graph, so it is necessary to parse it in order to find a sequence of productions that can generate such a graph. These productions can be next performed in the context of a new computing node (creating the new virtual machine).

4 Conclusion

The centralized control of the agents allocation is the bottle-neck of the presented solution, thus it is desirable introduction the parallel derivation of the distributed graph. Let us assume that each monolithic DCE maintains the local

graph and some information about the connections of this graph nodes with nodes belonging to the local graphs that are maintained by other DCEs. Modifications inside local graphs can be made in a parallel way. It is, also, easy to coordinate the common modification of two local graphs by implementing some communication protocol (basing for example on Two Commit protocol) with help of their monolithic DCEs and some systems **requests**. The presented solution, for ETPL(k) graph grammar [4] associated with DCE, offers polynomial parsing and generation complexity ($O(n^2)$) (where n is maximum of local graph dimensions (n_i) and number of local DCEs (k)). In case of the AI solutions, using multiple agents, we can expect that the influence of k parameter on a final system effectiveness shall grow. The computational complexity is here specially outlined, because most of the graph transformation mechanisms with large description power has an unsatisfactory computational complexity. For example, the most popular graph transformation mechanism based on single push-out has exponential computational complexity of parsing. This excludes the use of such a methodology in applications working on-line.

References

1. Ralph Depke, Reiko Heckel, and Jochen Malte. Integrating visual modeling of agent-based and object-oriented systems. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 82–83, New York, NY, USA, 2000. ACM Press.
2. H. Ehrig, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors. *Handbook of graph grammars and computing by graph transformation: vol. 3: concurrency, parallelism, and distribution*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1999.
3. H. Ehrig and G. Taentzer. Graphical representation and graph transformation. *ACM Comput. Surv.*, 31(3es):9, 1999.
4. M. Flasiński. Power properties of nlc graph grammars with a polynomial membership problem. *Theor. Comput. Sci.*, 201(1-2):189–231, 1998.
5. M. Flasiński and L. Kotulski. On the use of graph grammars for the control of a distributed software allocation. *The Computer Journal*, 35:A167–A175, 1992.
6. Mariusz Flasiński. Characteristics of ednlg-graph grammar for syntactic pattern recognition. *Comput. Vision Graph. Image Process.*, 47(1):1–21, 1989.
7. L. Kotulski. *Model wspomaganie generacji oprogramowania w rodowisku rozproszonym za pomoc gramatyk grafowych*. Rozprawy Habilitacyjne. Wydawnictwo Uniwersytetu Jagiellońskiego, 2000. ISBN 83-233-1391-1.
8. L. Kotulski and A. Nowak. Wykorzystanie gramatyk grafowych do kontroli migracji mobilnych agentów. In *Inżynieria wiedzy i systemy ekspertowe*, pages 319–325. Oficyna wydawnicza Politechniki Wrocławskiej, 2003.
9. Koenraad Mertens, Tom Holvoet, and Yolande Berbers. A case for adaptation of the distributed environment layout in multiagent applications. In *SELMAS '05: Proceedings of the fourth international workshop on Software engineering for large-scale multi-agent systems*, pages 1–8, New York, NY, USA, 2005. ACM Press.
10. Rong Xie, Daniela Rus, and Cliff Stein. Scheduling multi-task multi-agent systems. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 159–160, New York, NY, USA, 2001. ACM Press.

The Outline of the Strategy for Solving Knowledge Inconsistencies in a Process of Agents' Opinions Integration

Radoslaw Katarzyniak and Agnieszka Pieczyńska

Institute of Information Science and Engineering, Wrocław University of Technology,
Wybrzeże Wyspińskiego 27, 50-370 Wrocław, Poland
{radoslaw.katarzyniak, agnieszka.pieczynska}@pwr.wroc.pl

Abstract. In this paper a strategy for integration of agents' propositional attitudes is proposed in order to solve the semantic inconsistency between opinions of members of agents' population. In proposed algorithm all the agents' answers are transformed into fuzzy set equivalents and then final formula representing the agreement of all opinions is obtained.

1 Introduction

Lack of central mechanisms of control in multiagent systems caused that the states of some objects from external world might be for some agents not known. In such cases the agents must be equipped with mechanisms that let them cope with incomplete knowledge. In our approach this mechanism is called the algorithm of messages generation [4],[5]. As an input of this algorithm the overall agent's knowledge base is given and as an output the propositional attitudes (opinions) in which the content is related to the current states of objects. The agents' opinions are in the form of logic formulas [4],[9]. Each formula is consisted of the modal operator and two indices to the properties that are observed by the agent $a \in A$ in the object $o_z \in O$ at the time point $t_n \in T$ or are estimated on the basis of agent's previous experiences applying the algorithm for the messages generation. The co-existence of these properties in objects from the set O is reflected by logic connectives such as: conjunction, exclusive alternative or alternative. The language is given as: $\Omega = \{OP^a(P_i(o_z) \wedge P_j(o_z)), OP^a(P_i(o_z) \wedge \neg P_j(o_z)), OP^a(\neg P_i(o_z) \wedge P_j(o_z)), OP^a(\neg P_i(o_z) \wedge \neg P_j(o_z)), Bel^a(P_i(o_z) \vee P_j(o_z)), Bel^a(P_i(o_z) \vee P_j(o_z))\}$, where $OP = \{Bel, Pos, Know\}$. If the states of the objects are estimated on the basis of stored experiences then the modal operator of belief (Bel) or possibility (Pos) is used otherwise - the modal operator of knowledge (Know). For example for the formula $Pos^a(P_i(o_z) \wedge P_j(o_z))$ the spoken language interpretation is: In my (agent's a) opinion it is possible that the object o_z has the property P_i and the property P_j . Each modal operator has its numerical equivalent respectively: γ_{Know} , γ_{Bel} and γ_{Pos} that represents the level of agent's confidence. Additionally with each formula coefficient support α_p is correlated (see section 3). If an agent at the particular time point cannot perceive the current state of particular object from external world and the results of the algorithm for the message generation are

unsatisfied it asks other agents about their opinions. The agents send the answers that are the results of direct observation or applying the algorithm for messages generation. Due to the natural inconsistencies between prepositional attitudes carried out by different members of agents' population the incoming language responses can be different as regards to the used logic connectives in formulas and modal operators. Therefore, the agent interested in integration all these opinions, needs to apply certain strategies for semantic conflict resolution [5]. Proposed strategy is based on the idea of language grounding that states that each language formula needs to be related to a certain meaning embodied in private empirical experiences [1],[2],[3],[7].

2 A Strategy for the Integration of Conflicting Opinions

2.1 Basic Postulates

Postulates in relation to the states of objects:

1. If at the time point t_k the object o_z possesses both the property P_i and P_j then it belongs to the class of objects $O_1(t_k) \subseteq O$.
2. If at the time point t_k the object o_z possesses the property P_i and doesn't possess the property P_j then it belongs to the class of objects $O_2(t_k) \subseteq O$.
3. If at the time point t_k the object o_z doesn't possess the property P_i and possesses the property P_j then it belongs to the class of objects $O_3(t_k) \subseteq O$.
4. If at the time point t_k the object o_z doesn't possess neither the property P_i and P_j then it belongs to the class of objects $O_4(t_k) \subseteq O$.
5. If at the time point t_k the object o_z possesses either the property P_i or P_j then it belongs to the class of objects $O_5(t_k) \subseteq O$.
6. If at the time point t_k the object o_z possesses the property P_i or P_j then it belongs to the class of objects $O_6(t_k) \subseteq O$.

Postulates in relation to the messages:

1. With each message a support coefficient α_p , $p \in \{1, 2, \dots, 6\}$ is associated.
2. If an agent sends a message with conjunction as a logic connective then the support coefficient α_1 , $1=1, 2, 3, 4$ is equal 1. The state of an object o_z is unequivocally specified.
3. If an agent sends a message with exclusive alternative as a logic connective then the support coefficient α_5 is equal 0.5. The state of an object o_z is not unequivocally specified. Two states of affairs are considered: 1) an object o_z possesses the property P_i and doesn't possess the property P_j either 2) object o_z doesn't possess the property P_i and possesses the property P_j .
4. If an agent sends a message with alternative as a logic connective then the support coefficient α_6 is equal 0.3. The state of an object o_z is not unequivocally specified and three states of affairs are considered: 1) an object o_z possesses the property P_i and possesses the property P_j either 2) o_z possesses the property P_i and doesn't possess the property P_j either 3) o_z doesn't possess the property P_i and possesses the property P_j .

5. Each message φ_a about the state of an object o_z can be transformed into its first-level $fs^1_a(o_z)$ and second-level $fs^2_a(o_z)$, $a=\{1,2,\dots,\text{card}(A')\}$ semantically equivalent fuzzy set representation. First-level fuzzy set representation $fs^1_a(o_z)$ represents only the state of an object in relation to the co-existence of the properties P_i and P_j in the object o_z , but in the second-level representation $fs^2_a(o_z)$ also the power of agent's beliefs is taken into consideration.
6. If two first-level fuzzy set representations $fs^1_a(o_z)$ and $fs^1_b(o_z)$ consist a support coefficient α_i , $i \in \{1,2,3,4\}$ then the support levels for the correlated formulas φ_a and φ_b as a consensus opinion about the state of an object o_z are mutually equal.
7. If there is a support coefficient α_i , $i \in \{1,2,3,4\}$ in the first-level fuzzy set representation $fs^1_a(o_z)$ and α_5 (or α_6) in the first-level fuzzy set representation $fs^1_b(o_z)$, then the support level for the formula φ_a as a consensus opinion about the state of an object o_z is higher than the one for the formula φ_b .
8. If there is a support coefficient α_5 in the first-level fuzzy set representation $fs^1_a(o_z)$ and α_6 in the first-level fuzzy set representation $fs^1_b(o_z)$, then the support level for the formula φ_a as a consensus opinion about the state of an object o_z is higher than the one for the formula φ_b .

2.2 The Idea of the Algorithm for the Integration Propositional Attitudes

Input: The set of answers $Y=\{\varphi_1, \varphi_2, \dots, \varphi_{\text{card}(A')}\}$ sent by the agents from $A' \subseteq A$.

Output: The opinion φ_C about the state of an object o_z in relation to the properties P_i and P_j with the confidence coefficient γ_C .

Parameters: $\gamma_{\text{Know}}=1, \gamma_{\text{Bel}}=0.7, \gamma_{\text{Pos}}=0.4, \delta$ - the accepted value of tolerance in relation to the support function's value.

Step 1. Transform each formula φ_a into its first-level fuzzy set equivalent fs^1_a as follows:

- if $\varphi_a = OP^a(P_i(o_z) \wedge P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 1), (k_2, 0), (k_3, 0), (k_4, 0)\}^\gamma$
- if $\varphi_a = OP^a(P_i(o_z) \wedge \neg P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 0), (k_2, 1), (k_3, 0), (k_4, 0)\}^\gamma$
- if $\varphi_a = OP^a(\neg P_i(o_z) \wedge P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, 1), (k_4, 0)\}^\gamma$
- if $\varphi_a = OP^a(\neg P_i(o_z) \wedge \neg P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, 0), (k_4, 1)\}^\gamma$
- if $\varphi_a = OP^a(P_i(o_z) \vee P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0.5), (k_3, 0.5), (k_4, 0)\}^\gamma$
- if $\varphi_a = OP^a(P_i(o_z) \vee \neg P_j(o_z))$ then $fs^1_a(o_z) = \{(k_1, 0.3), (k_2, 0.3), (k_3, 0.3), (k_4, 0)\}^\gamma$

where $OP \in \{Know, Bel, Pos\}$, γ represents the level of agent's confidence and is correlated with modal operators as follows: if $OP=Know$ then $\gamma=\gamma_{\text{know}}=1$, if $OP=Bel$ then $\gamma=\gamma_{\text{Bel}}=0.7$, if $OP=Pos$ then $\gamma=\gamma_{\text{Pos}}=0.4$

Step 2. Transform each first-level fuzzy set representation $fs^1_a(o_z)$ into second-level fuzzy set representation $fs^2_a(o_z)$ as follows:

- if $fs^1_a(o_z) = \{(k_1, 1), (k_2, 0), (k_3, 0), (k_4, 0)\}^\gamma$ then $fs^2_a(o_z) = \{(k_1, f_1(1, \gamma)), (k_2, 0), (k_3, 0), (k_4, 0)\}$
- if $fs^1_a(o_z) = \{(k_1, 0), (k_2, 1), (k_3, 0), (k_4, 0)\}^\gamma$ then $fs^2_a(o_z) = \{(k_1, 0), (k_2, f_2(1, \gamma)), (k_3, 0), (k_4, 0)\}$
- if $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, 1), (k_4, 0)\}^\gamma$ then $fs^2_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, f_3(1, \gamma)), (k_4, 0)\}$
- if $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, 0), (k_4, 1)\}^\gamma$ then $fs^2_a(o_z) = \{(k_1, 0), (k_2, 0), (k_3, 0), (k_4, f_4(1, \gamma))\}$
- if $fs^1_a(o_z) = \{(k_1, 0), (k_2, 0.5), (k_3, 0.5), (k_4, 0)\}^\gamma$ then $fs^2_a = \{(k_1, 0), (k_2, f_5(0.5, \gamma)), (k_3, f_5(0.5, \gamma)), (k_4, 0)\}$

if $fs_a^1(o_z) = \{(k_1, 0.3), (k_2, 0.3), (k_3, 0.3), (k_4, 0)\}^\gamma$ then $fs_a^2(o_z) = \{(k_1, f_6(0.3, \gamma)), (k_2, f_6(0.3, \gamma)), (k_3, f_6(0.3, \gamma)), (k_4, 0)\}$, where f_p is the membership function of the object $o_z \in O$ to the class of objects $O_p(t_k)$, $p \in \{1, 2, \dots, 6\}$ and is defined as: $f_p: \{1, 0.5, 0.3\} \times \{1, 0.7, 0.4\} \rightarrow (0, 1)$

Remark: For each $fs_a^1(o_z)$ only one value δ_p of the membership function f_p , $p \in \{1, 2, \dots, 6\}$ is computed; $f_p(\alpha_p, \gamma) = \alpha_p \cdot \gamma$

Step 3. Compute fuzzy set representative $fsr^2(o_z)$ of all second-level fuzzy set representations $fs_a^2(o_z)$, $a = 1, 2, \dots, \text{card}(A')$: $fsr^2(o_z) = \{(k_1, \Theta_1), (k_2, \Theta_2), (k_3, \Theta_3), (k_4, \Theta_4)\}$,

$$\Theta_1 = \frac{1}{\text{card}(A')} \sum_{\substack{fs_a^2(o_z) \in Fs^2(o_z) \\ p \in \{1, 6\}}} f_p(\alpha_p, \gamma) \quad \Theta_2 = \frac{1}{\text{card}(A')} \sum_{\substack{fs_a^2(o_z) \in Fs^2(o_z) \\ p \in \{2, 5, 6\}}} f_p(\alpha_p, \gamma)$$

$$\Theta_3 = \frac{1}{\text{card}(A')} \sum_{\substack{fs_a^2(o_z) \in Fs^2(o_z) \\ p \in \{3, 5, 6\}}} f_p(\alpha_p, \gamma) \quad \Theta_4 = \frac{1}{\text{card}(A')} \sum_{fs_a^2(o_z) \in Fs^2(o_z)} f_4(\alpha_p, \gamma)$$

$Fs^2(o_z)$ is the set of all second-level fuzzy set representations.

Step 4. Apply the procedure of the choice of consensus description of the state of object o_z [see 5].

Stop.

3 Final Remarks

In this paper a strategy of integration of agents' propositional attitudes was proposed in order to solve the semantic inconsistency between agents' opinions. The receiver of the messages applies the algorithm for the propositional attitudes integration. In this algorithm all the agents' answers are transformed into fuzzy set equivalents and then the formula representing the agreement of all opinions is obtained.

References

1. Coradeschi S., Saffiotti A., An Introduction to the Anchoring Problem, Robotics and Autonomous Systems 43, (2003), 85-96
2. Harnad, S.:The Symbol Grounding Problem. Physica, 42, 335-236
3. Katarzynyak, R., Pieczyńska-Kuchtiak, A.: A Consensus Based Algorithm for Grounding Belief formulas in Internally Stored Perceptions. Neural Network World, 5, (2002) 671-682
4. Katarzynyak, R., Pieczyńska-Kuchtiak, A.: Grounding and extracting modal responses in cognitive agents: AND query and states of incomplete knowledge. International Journal of Applied Mathematics and Computer Science, 14(2), (2004), 249-263.
5. Katarzynyak, R., Pieczyńska-Kuchtiak, A.: An Approach to resolving semantic inconsistency of multiple propositional attitudes. Journal of Intelligent & Fuzzy Systems 17(3), (2006), (to appear).
6. Pieczyńska-Kuchtiak, A., Towards measure of semantic correlation between messages in multiagent system, ICCS 2005, Lecture Notes on Computer Science, Kraków, (2004) pp. 567-574.
7. Vogt P., Anchoring of semiotics symbols, Robotics and Autonomous Systems, 43, (2003) 109-120.

Agent-Based Service Discovery Middleware in Ubiquitous Environments*

Hyung-Jun Kim, Kyu Min Lee, Kee-Hyun Choi, and Dong Ryeol Shin

School of Information & Communication Sungkyunkwan University,
Jangangu Chunchundong 300, Suwon Kyunggido, Korea
{mairi, kmlee, gyunee, drshin}@ece.skku.ac.kr

Abstract. Service discovery is becoming a hot topic as it becomes more widespread through the Internet. In pervasive environments, there are various services, and users use a service discovery protocol for finding the most appropriate service among them. Thus, over the past few years, many service discovery protocols have been proposed. Even though they provide convenient service discovery functionality with users, complexity is increased because they have different message exchange pattern and architecture. In this paper, we propose a novel middleware based on agent platform for interoperability among heterogeneous services. This has the Discovery Agent for each service discovery protocols and it can detect service advertisement messages for registration, so the services are registered in agent platform. Finally, user can use them. The key point of this is not to require modification of existing service discovery protocols. Near the conclusion of this paper, this is implemented.

1 Introduction

Over the past few years, lots of service discovery protocols have been proposed. Although they provide the same basic functionality of service discovery, they differ significantly in architecture, message exchange pattern, expected operating environment and service representation/description. These differences make they cannot discover services of other service discovery protocols. In this paper, we propose a new middleware that provides agent-based service discovery to support interoperability using a FIPA-compliant agent platform [1]. The proposed middleware is different from the currently available approaches in that users are easily able to use heterogeneous services through activating discovery agent. It does not require modifying existing service discovery protocols, and uses JADE (Java Agent Development Framework) among diverse agent platforms, which is representative and popular, so it is simple to develop an agent. Other benefits are not only interoperability, but also agent collaboration. The rest of this paper is organized as follows. Section 2 presents the proposed architecture. Section 3 describes operation of our system. In section 4, the implementation is presented and this paper is finally concluded in section 5.

* This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

2 Proposed Architecture Overview

Fig. 1 shows our new middleware architecture.

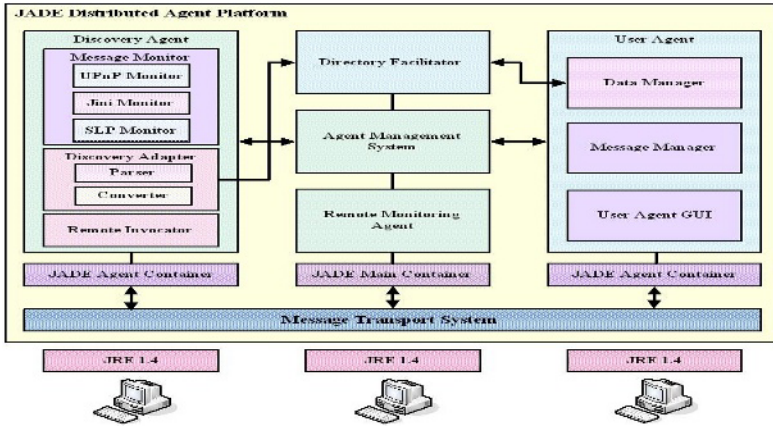


Fig. 1. Architecture of our middleware

As illustrated in Fig. 1, there are three parts on the middleware.

Discovery Agent

Discovery Agent receives messages from heterogeneous service discovery protocols through each well-known message port. In other words, there is Message Monitor module for the function, and JADE platform is able to receive service registration messages of UPnP (Universal Plug and Play) [2] and SLP (Service Location Protocol) [3] domain. These messages are converted to DAD used on JADE after parsing and converting process through Discovery Adaptor module.

In fact, Discovery Adaptor analyzes messages and then calls appropriate parser and converter. Namely, if the messages are from UPnP domain, it uses the parser and converter for UPnP. Otherwise, if the messages are from SLP domain, it uses the parser and converter for SLP domain. After conversion, newly created services of DAD type are automatically registered in DF. Remote Invocator is a module that invokes the service User Agent wants to use. For example, it creates SOAP message for invocation, the user agent wants to invoke an UPnP service.

JADE Main Container

If users start up JADE platform, main container is basically created, and DF agent, which is a yellow page, AMS, which is a white page, and Remote Management Agent, which is a JADE management tool, are created and executed. DF stores heterogeneous services that are detected by Discovery Agent. Finally, User Agent can use them by searching DF. All agents on JADE should get AID (Agent Identity Description) from AMS and then is executed.

User Agent

User Agent gets information of UPnP and SLP services that are registered in DF by polling method. It can discover services through the process. Message Manager supports communication with other agents and when an agent is created, a user can see the process on GUI (Graphic User Interface).

3 Sequence Diagram

Fig. 2 shows a sequence diagram of using services of UPnP domain and SLP domain. First, UPnP device and Service Agent of SLP advertise their services. Discovery Agent detects each registry message through well-known ports, parses them and converts them to DAD type. Next, it registers them in DF. After that, User Agent can get the list of heterogeneous services that are in DF. If we want to use them, we can send ACL message to Discovery Agent. Finally, Discovery Agent parses the message, and then creates SOAP message to invoke the services.

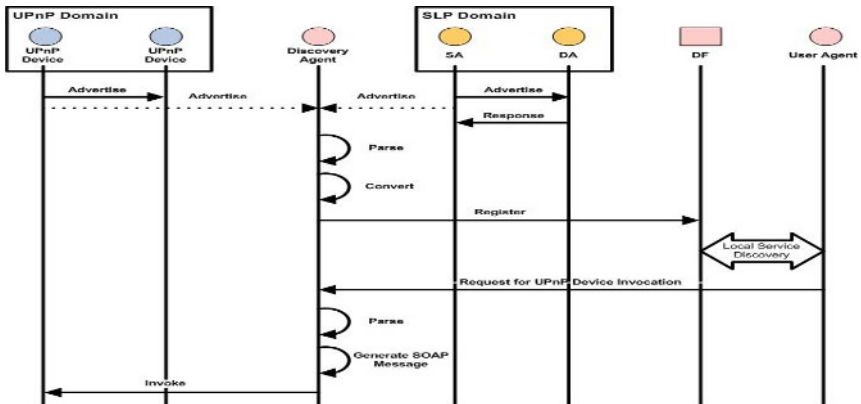


Fig. 2. Sequence diagram: Service Registration and Discovery

4 Implementation and Application

In this Section we demonstrate the implementation of the new middleware. Implementation of our middleware is based on JADE 3.3 and each agent is created on 1.4.2_07 version.

The prototyped system is based on UPnP development tool with Intel's UPnP SDK, an open source implementation of the UPnP protocol for UNIX, and Siemen's Java UPnP stack, and SLP development using mSLP supported by Columbia Univ.

We did test DVD Shop, Bank and Hotel services on mSLP and TV device, Washer Device, and Light device on UPnP.

As can be seen in Fig. 3, it shows Discovery Agent and User Agent are on JADE where User Agent indicates a user who tries to find the services in DF.

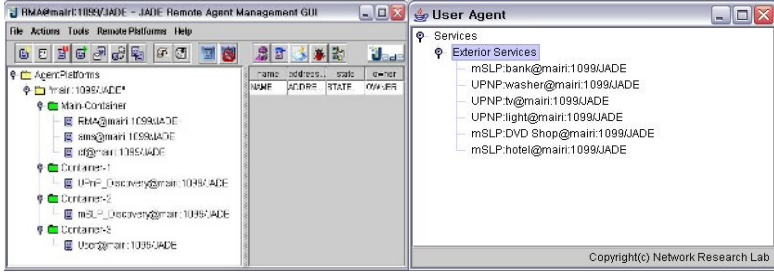


Fig. 3. JADE RMA and User Agent GUI

5 Conclusion

Existing service discovery protocols have shortcoming because they cannot guarantee interoperability among them. To solve the problem, we propose novel middleware based on agent platform, which supports scalability and generality

Until now, we have implemented our middleware, UPnP and SLP Discovery Agent. In the future, we will develop Discovery Agent for Jini. The most important thing is the middleware supports users to be able to use heterogeneous services in ubiquitous environments without any modification of existing service discovery protocols.

References

- [1] Foundation for Intelligent Physical Agents. *FIPA Agent Management Specification*. June 2002.
- [2] UPnP Forum, "UPnP device architecture 1.0," May 2003.
- [3] E. Guttman, C. Perkins, J. Veizades, and M. Day, "RFC 2608: Service location protocol, version 2," 1999, status: PROPOSED STANDARD.
- [4] M. Rajdou. Software agents in business: Steady adoption curve. Technical report, Forrester Research, USA, 2003.
- [5] Michael Wooldridge. *An Introduction to MultiAgent System*. Wiley and Sons, 2002.
- [6] Michael Luck, Peter McBurney, and Chris Preist. *Agent Technology: Enabling Next Generation Computing*. <http://www.agentlink.org/roadmap>, 2003.

An Intelligent Middleware Architecture for Context-Aware Service Discovery*

Kyu Min Lee, Hyung-Jun Kim, Kee-Hyun Choi, and Dong-Ryeol Shin

School of Information and Communication Engineering,
Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{kmllee, mairi, gyunee, drshin}@ece.skku.ac.kr

Abstract. There is a high interest on service discovery techniques, which minimize the cost of detecting services and provide users with convenience. For more dynamic and useful service discovery, middleware for context-aware service discovery is required. This paper proposes an intelligent middleware architecture for context-aware service discovery that is based on JADE, which is a FIPA-compliant agent platform. The proposed middleware provides not only service scalability among heterogeneous domains, but also modules processing context information. When a service is detected, context information relating to the user and environment is used. As a policy-based system, the proposed middleware also use predefined policy. In other words, user preference can be considered. Near the conclusion of this paper, we discuss future works.

1 Introduction

Both service providers and users of these services require a middleware system to integrate heterogeneous environments, employing various techniques to reduce complexity. There are many existing middleware system to solve that problems. Although they try to solve integration problems, another problem emerges. That is to use context information such as temperature, position and body information because service users can find more appropriate service if the middleware reflects context information, so recent middleware approach is to support context-awareness.

In this paper, we propose a middleware architecture based on Java Agent DEvelopment Framework (JADE) [1], a popular agent platform, for context-aware service discovery. In the proposed middleware, raw context information is discarded and useful raw data is represented to high-level data. Context information can be combined, inferred by Context Manger which is comprised of several modules in our middleware. Because our middleware also have Policy Module, it can consider users' preference.

The remainder of the paper is organized as follows. Section 2 presents our proposed middleware for context-aware service discovery in detail. Finally, the conclusion is discussed in section 3.

* This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

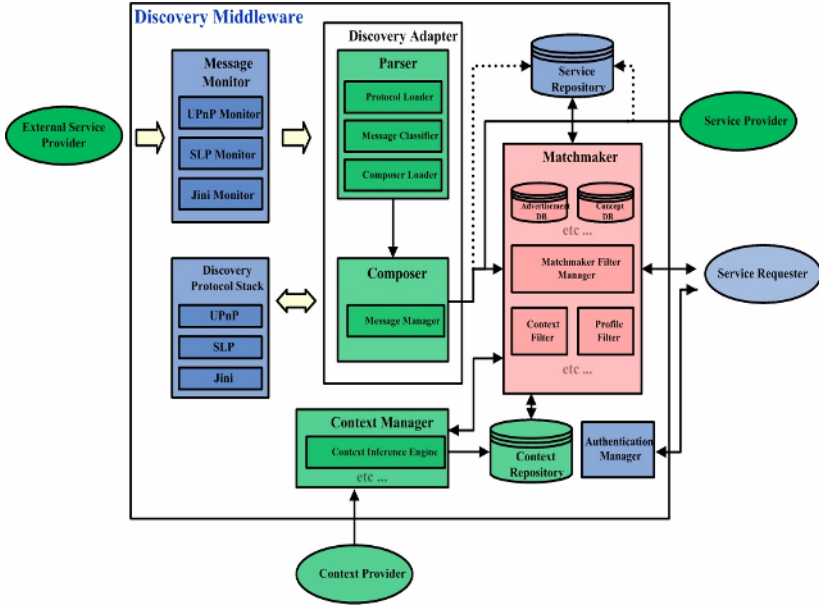


Fig. 1. The Architecture of Middleware

2 Proposed Middleware System

2.1 Architecture

Novel middleware is designed for context-aware service discovery. As demonstrated in Figure 1, this middleware consists of various modules such as parser, composer, service repository, and so on. In this paper, context-awareness in middleware is focused on, therefore other modules are discussed only briefly.

The Message Monitor has each discovery agent as a monitoring module for each service discovery protocol or network domain that can detect all messages from them because each network protocol or domain uses a well-known IP and Port number, i.e. 239.255.255.250:1900 on UPnP and 239.255.255.253:2427 on mSLP. The Discovery Adapter consists of a Parser, which parses messages, and Composer, which composes indispensable data parsed from messages. It means the Composer represents the parsed data to the appropriate format, DAD (DF-Agent-Description) [1] format. The Composer then registers it with the Service Repository such as the Directory Facilitator (DF) in an agent platform [1]. The agent-based service provider can also register services directly via an Agent Communication Language (ACL) message [1]. When a service requester on behalf of a user requests a service, a request message is transmitted to the Matchmaker, which calculates which service is the most appropriate for the service request. The Context Provider such as light, camera and temperature, serves raw context information to the Context Manager. The Context Manager consists of modules which guarantee context-awareness. Useful context information is stored in

the Context Repository. If you want to see detail information about the middleware, you can see [2].

2.2 Context Awareness

By introducing modules in Context Manager, people can know how Context Manager is comprised of. The Context Manager consists of six modules and one database as shown in Figure 2. Each of them can be described as the following:

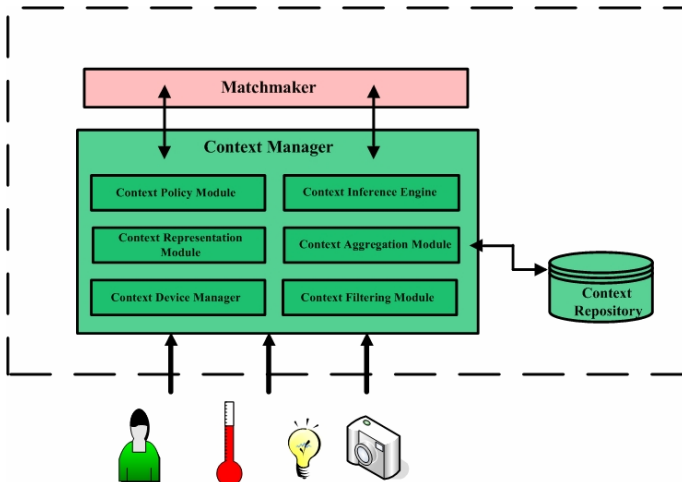


Fig. 2. Context Modules

- **Context Device Manager:** In ubiquitous environments, there are many diverse context devices and context sources. They frequently enter or go out in the network, so we design the Context Device Manager, which manages them. When they register or deregister, a message is sent to the Context Device Manager. Then, it recognizes the context sources composed in the middleware network.
- **Context Filtering Module:** The next module is the Context Filtering Module. Its main purpose is to protect the Context Repository from being flooded with excessive information. When context information (e.g., time) from context sources is continuously delivered, users or user agents may be interested in receiving a few values periodically. This is especially important for both the Matchmaker and users because the Matchmaker can search requested context information in the Context Repository as soon as possible and users reduce the response time of the Matchmaker.
- **Context Representation Module:** The Context Representation Module converts raw context information from context sources such as sensors into normalized forms. Information from various context sources, in general, exist in ubiquitous computing environments and there are many different types of data, e.g., binary, integer, real number, etc. For the more, the representation may be different from each other. This will result in difficulties not only in usability, but also in production of higher-level contexts from lower-level contexts. This module represents raw data in normalized forms.

- **Context Aggregation Module:** The Context Aggregation Module has the ability to combine the context information from the Context Representation Module. This module produces combined contexts. The combined context can be used to provide better information with users because they can understand easily what kind of situation. For example, when Context Representation Module generates Location (“713”), Patient (“Chris”), Sensor (“Bodily Temperature”), Temperature (“34”), Context Aggregation Module creates Room(Number, “713”, Patient, “Chris”, Sensor, “Bodily Temperature”, Temperature, “34”).
- **Context Inference Module:** Context Inference Module can contain one or more context reasoning modules, based on the users’ requirements. The need for a Context Inference Module arises because not all information can be gathered from context sources. It is used to derive higher-level context information from lower-level information. Therefore, the module infers new context information from the current context. This makes the middleware system increasingly intelligent. For example, in a room a man lies down on a bed, he turns off lights and closes his eyes. This means he is sleeping.
- **Context Policy Module:** When a situation or a environment is configured, users might want to know the moment. At that time, middleware invokes actions already described. This is possible due to the Context Policy Module. User can send policy message as an ACL message format to the Context Agent, which has Context Policy Module. In other words, this invokes actions if a pre-defined context is detected in the current situations or surroundings.

3 Conclusion

This paper proposes middleware architecture based on JADE for context-aware service discovery. We are currently implementing the middleware in accordance with a hospital scenario because the best way to test this is to create a scenario.

References

1. JADE, <http://jade.tilab.com>, Telecom Italia Lab, 2005.
2. Hyung-Jun Kim, Kyu Min Lee, Kee-Hyun Choi and Dong-Ryeol Shin, “Service Discovery using FIPA-Compliant AP to Support Scalability in Ubiquitous Environments”, Proc. of the IEEE on ICIS, Pages: 647-652, July 2005.

Mobile Agent Based Publication Alerting System

Ozgur Koray Sahingoz¹ and A. Coskun Sonmez²

¹ Air Force Academy, Computer Engineering Department, Yesilyurt, Istanbul, Turkey
sahingoz@hho.edu.tr

² Yildiz Technical University, Computer Engineering Department, Yildiz, 34349,
Istanbul, Turkey
acsonmez@ce.yildiz.edu.tr

Abstract. This paper introduces a distributed publication alerting service which increases the amount of information in notification message while using information hiding principle. It is aimed to design a mobile agent based publication alerting system (MABPAS) which uses mobile agents to dispatch notification information (generally all text information) about produced publication to subscribers. MABPAS combines the advantages of publish/subscribe communication model and mobile agents into a flexible and extensible distributed execution environment.

1 Introduction

Publication alert is an interactive feature that makes it possible for authors and publishers to directly post notification to an information dissemination system (like mail groups or dispatching services) about recently published works. This approach makes subscribers of the system aware of recent publications. To alert more subscribers it is necessary to put more information to notification message.

Publication alerting systems are used to provide online platform for keeping subscribers to be informed about any publication in a new issue of a journal, magazine, newsletter or proceedings. Publications can be either scientific publications like technical reports, conference papers, journal papers, books or social publications like novels, adventure books and magazines. In most of the previous works, notifications about publications are performed by mailing lists with limited expressiveness and functionality. To improve the efficiency of the alerting system we developed a mobile agent based publication alerting service (MABPAS) which takes advantage of mobile agents to dispatch necessary information (generally all text information) about produced publication to subscribers. MABPAS uses the execution environment of an agent based distributed event system [1] which uses mobile agents as event messages (called as *agvent(agent event)*).

2 Publication Alerting System

MABPAS implements the publish/subscribe protocol, thus enabling many-to-many interaction of loosely coupled entities. It also allows publishers and subscribers to

dynamically connect and disconnect from the system, a capability that extends the flexibility of the working environment. Two properties of MABPAS make it distinctive from other alerting systems:

- *Autonomy*: the agent structure of the publication agents represents both the capabilities (ability to compute something, to decide target nodes) and the preferences over how these capabilities can be used. Thus, publication agents have the ability to reason about how they use their resources and selects target subscribers by using knowledgebase of the brokers. These brokers are called as *agent servers* whose main role is providing an execution platform for publication agents.
- *Conversation-Negotiation*: since the agents are autonomous, they must negotiate with other stationary agents to gain access to their resources and capabilities. It is important to hide information in the agent structure; therefore both agent servers(ASs) and subscribers cannot see information in it. At the same time, agents cannot reach directly to the resources of ASs. Agents can obtain necessary information for their routing operations by talking with stationary agent of an AS. Communication with the subscriber is a little bit different. There is no forwarding operation after arriving on a subscriber. Therefore a publication agent does not need to talk with the subscriber’s stationary agent. However a subscriber agent has to communicate with the incoming publication agent to get necessary information. To accomplish this subscriber agent follows a "conversation sequence", where multiple messages are exchanged according to some pre-defined protocols.

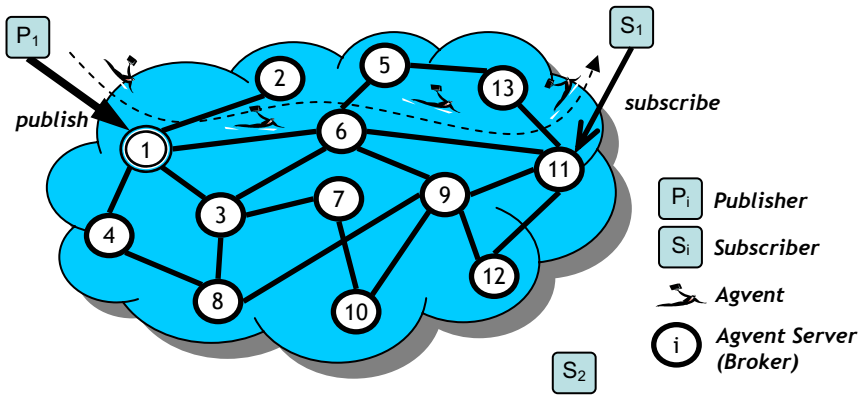


Fig. 1. Dispatching Mechanism of MABPAS

MABPAS consists of four main components as depicted in Figure 1;

1. *Publishers* are components which produce publications and publish their relevant agents to the alerting system (i.e. publishing houses or singular authors).
2. *Subscribers* are the consumers of the system like libraries, researchers, academic staff or students. They issue subscriptions/profiles that describe the publications/agents they want to receive.
3. *Agents* are created by publishers according to relevant publications and they are the main components of the system. A publication agent is a collection of code and

data (generally all text information of the publication) that migrates through the dispatch service. Dispatch service is comprised of a network of distributed nodes and an agent routes itself at each node on the path towards its target subscribers. The agent structure of MABPAS is named as *PublicationAgent* which contains attributes and behaviors of the relevant publication. These properties are used for establishing an efficient filtering mechanism. *PublicationAgent* is a class definition which contains not only the filterable attributes like *publisher_name*, *publish_date*, *pages*, *authors*, *publication_name*, *list_price...etc* but also filterable methods like *ReferenceContains(String AuthorName)*, *TOCContains(String topic)*, *WholesalePrice(int amount,String destination)...etc*. The filtering process of the system is the keystone of the alerting service. System enables subscribers to subscribe on both the publication agents' attributes and methods. By using this type filtering mechanism; it is possible to compose complex subscriptions, message traffic of system is reduced, and dispatching of irrelevant messages is prevented.

There are also some methods which are used for conversation with subscribers. These methods are not filterable methods and therefore they are not advertised. These methods are used by publication agent according to subscribers' requests.

4. *Agent Server* processes incoming subscriptions and advertisements according to a protocol which includes their propagation to adjacent/neighbor ASs. Agents are self-routing, that is, they are responsible for determining their own paths through the network, utilizing a minimal set of facilities provided by ASs. ASs support incoming agents by providing a simple, architecturally independent environment for the receipt and execution of publication agents. An AS has no access to the content of the published agent, which simplifies its role and consequently facilitates the server development process. Information hiding principle also meets confidentiality of publication information.

3 Performance Evaluation

We've selected average publication (agent) distribution time as an appropriate benchmark for evaluating performance of MABPAS. To calculate this we use a network with properties as shown in Table 1.a and we dispatch a publication agent with the properties as shown in Table 1.b.

Table 1. Model Parameters for Performance Evaluation

Properties	Explanation
LAN	100 Mbit Ethernet
LAN Op. Sys	Windows NT
Computer Properties	Pentium III, IV, Centrino 1.0, 3.0, 1.6 GHz CPU 512 Mb RAM 256, 512, 2000 Kb Cache 133, 800, 400 MHz FSB Windows XP Op. Sys.

a) Network Properties

Properties	Value Space
# of advertisements	5,000-15,000
# of subscriptions	10,000-30,000
# of neighbor agent servers	1-7
# of subs. connected to an AS	5-100
# of ASs in dispatching system	100
# of attributes in subscription	0-15
# of behaviors in subs.	0-10
# of parameters for method call	0-5

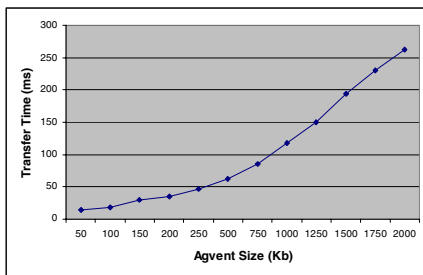
b) System Properties

We define distribution time as the elapsed time from the generation of a publication agent to its reception by all subscribers that are interested in that agent type. In our analysis, agent generation is the submission of a publication agent to an AS. In our preliminary analysis, we established an objective of determining the efficiency of the AS delivery without incurring concurrency into the system. Our goal was to examine the performance and scalability of the alerting system.

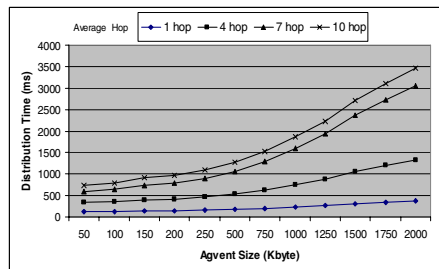
Table 2. Agent Sizes

Type of Publication	# of Word	# of Characters	Approximate Agent Size (Kb)
Article in a Newspaper	641	4301	4.60
Article in a Magazine	1943	12658	13.93
Paper with 8 page	3116	17260	22.35
Paper with 12 page	7504	40660	53.81
PHD Thesis	44650	318920	320.17
Symposium Proceedings With 50 Papers	237300	1308750	1,700.00
Book with 870 Pages (i.e. Comp. Netw. Tanenbaum)	325070	1600654	2,330.00

Dispatching of an agent is directly related with its size. It changes according to agent’s type and the data it contains. Table 2 shows the change in agent size relative to number of words in a publication.



a) Transfer time between two ASs



b) Agent Distribution Time

Fig. 2. Experimental Results for Agent Transfer Times

To compile realistic network topologies that approximate the behavior of real wide-area networks, we used a generator of random network graphs that implements the Transit-Stub model [2] with 100 agent servers. We ran a simple benchmark with a single publisher and a multiple number of subscribers, which were distributed evenly on this alerting service. As we were sure that there was no concurrent agent delivery, we could compute the theoretical best average delivery time. Figure 2.a shows agent transfer times between two ASs relative to agent sizes. Figure 2.b shows agent distribution time in MABPAS with previously mentioned system parameters for different average hops between publishers and subscribers.

References

1. Sahingoz, O. K., and Erdogan N.:Agent Based Distributed Event System, Proceedings of 30th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2004), Czech Republic, (2004), 144-153.
2. Zegura, E. W., Calvert, K. L., and Bhattacharjee, S.: How to Model an Internetwork. In Proceedings of IEEE INFOCOM '96, San Francisco CA, U.S.A., (1996), 594-602

Maintaining Diversity in Agent-Based Evolutionary Computation

Rafał Dreżewski and Marek Kisiel-Dorohinicki

Department of Computer Science,
AGH University of Science and Technology, Kraków, Poland
{drezew, doroh}@agh.edu.pl

Abstract. Niching techniques for evolutionary algorithms are aimed at maintaining the diversity through forming subpopulations (species) in multi-modal domains. Similar techniques may be applied to evolutionary multi-agent systems, which provide a decentralised model of evolution. In this paper a specific EMAS realisation is presented, in which the new species formation occurs as a result of co-evolutionary interactions between preexisting species. Experimental results aim at comparing the approach with a classical niching techniques and a basic EMAS implementation.

1 Introduction

The term Evolutionary Computation is usually used to describe a wide range of *global* search and optimization techniques based on analogies to natural evolutionary processes. However, both experiments and analysis show that for multi-modal problem landscapes a simple evolutionary algorithm will inevitably lose the diversity of its population and in consequence locate only a single solution—which is often a *local* optimum. In order to overcome this limitation some mechanism that creates and maintains different subpopulations (*species*) in a multi-modal domain must be used.

Traditionally, there are two basic approaches in evolutionary biology to understanding *speciation process* [3]. The first one called *allopatric speciation* occurs when subpopulations of a given species become geographically isolated. After isolation they follow different paths of evolution, which eventually lead to forming of different species. The second kind of speciation is called *sympatric speciation*. Such speciation results from niche separation due to resource competition, predator-prey co-evolution, sexual selection, etc. In this case speciation process takes place within single population.

Evolutionary multi-agent systems assume a *decentralised* model of evolution [5], which incorporates mechanisms of both allopatric (geographical isolation due to physical distribution of subpopulations) and sympatric (competition for limited resources in energy-based selection) speciation. In fact, these mechanisms prove not powerful enough to maintain stable subpopulations locating different optima. That is why the idea of *co-evolutionary* multi-agent system (CoEMAS) was introduced [2], which opens new possibilities of modeling biological speciation mechanisms based on co-evolutionary interactions—like predator-prey interactions, sexual preferences, etc.

2 Maintaining Diversity in Evolutionary Algorithms

During the years of research various mechanisms and techniques for creating and maintaining species located within the basins of attraction of local minima (niches) have been proposed. All these *nicheing techniques* allow species formation via the modification of the parent selection mechanism (*fitness sharing* [4]), the modification of mechanism of selecting individuals for new generation (*crowding* [6]), the restriction of application of the selection and/or recombination mechanisms (by *grouping* individuals into subpopulations or by introducing the environment with some topography in which the individuals are located [1]).

In *co-evolutionary algorithms* the fitness of each individual depends not only on the quality of solution to the given problem but also (or solely) on other individuals' fitness. This makes such techniques applicable in the cases where the fitness function formulation is difficult (or even impossible). Co-evolution of species was also used as a mechanism for maintaining population diversity, introducing open-ended evolution and improving adaptive capabilities of evolutionary algorithms in dynamic environments. As the result of ongoing research quite many co-evolutionary techniques have been proposed. Generally, each of these techniques belongs to one of two classes: competitive [7] or cooperative [8].

3 Evolutionary and Co-evolutionary Multi-agent Systems

The main idea of *evolutionary multi-agent system* is the modeling of evolution process in MAS [5]. In opposition to classical evolutionary algorithms, in *EMAS* there is no centralized algorithm which manipulates the whole population. All individuals (*agents*) are independent and make their own decisions, particularly these concerning reproduction and death.

Selection in *EMAS* is based on a non-renewable resources, which are possessed by the agents. Every activity costs some resource and may be realized provided that the agent's resource level is high enough. Resources can be gained only from the environment or other agents, and the rule is that better fit agents are given more resources than less fit ones. This means the competition for limited resources in common environment, which allows for *sympatric speciation*.

Agents live within an environment with a defined spatial structure and may change their location within the environment. It is thus a natural model of *allopatric speciation* because evolutionary processes may run with different conditions in many locations co-existing in parallel. Agents migrate, so the genetic material (information) is exchanged between locations. This model is, to some extent, similar to *island model* of PEA with all its advantages and disadvantages.

The key idea that differentiates *co-evolutionary* multi-agent system (CoEMAS) from *EMAS* is the *co-evolution* of several species (which cooperate or compete) in one environment [2]. In the particular model of *EMAS* with *co-evolving species* (NCoEMAS) there is a hierarchical structure of species: agents-solutions live within agents-niches.

In NCoEMAS there is a closed circulation of resources. Some resource is given from the environment to the agents-niches (proportionally to their fitness) and then

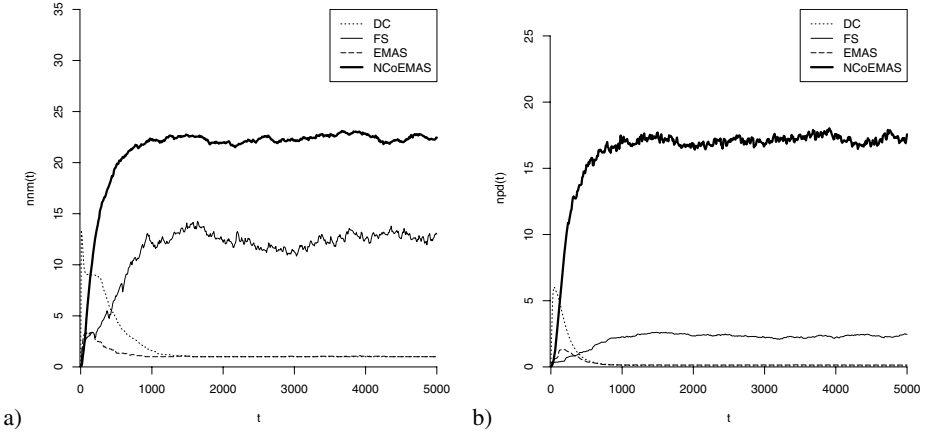


Fig. 1. The number of Rastrigin’s function local minima neighborhoods located (a) and the value of proportional species’ sizes indicator (b) – average values from 20 runs of different algorithms

redistributed among agents-solutions (a competition for limited resources). In spite of the proportional amount of resource, all agents-niches are also given some minimal amount of resource in order to keep alive less fitted species of solutions. The resource spent by the agents returns to the environment.

Each time step every agent-solution searches for the agent-niche, which is located within the basin of attraction of the same local minimum. If there are no such agents-niches, the agent-solution creates a new agent-niche, which genotype is the copy of its own genotype and migrates into it. Then each agent-solution, which have enough resource, searches its agent-niche for the reproduction partner.

In the considered case genotypes of all agents are real-valued vectors. Intermediate recombination and mutation with self-adaptation are used for agents-solutions and special mutation for agents-niches (the resulting genotype is the center of gravity of agents-solutions that belong to the agent-niche which genotype is mutated). Only agents-niches can migrate within the environment. They can also merge if they are located at the same place of environment and if they are located within the basin of attraction of the same local minimum.

4 Experimental Results

The experiments were carried out for four techniques: EMAS, and NCoEMAS, as well as deterministic crowding (DC [6]) and fitness sharing (FS [4]). Figure 1a shows the average number of Rastrigin’s function local minima neighborhoods located by the compared systems. The local minima neighborhood was classified as “located” when there was at least 3 individuals closer than 0.05 to that local minima. Figure 1b shows the average values of proportional species’ sizes indicator $npd(t)$. The $npd(t)$ indicator gives the higher measure when the sizes of species located within basins of attractions of local minima are proportional to the “quality” of that local minima.

The results of experiments indicate that *NCoEMAS* located more neighborhoods of local minima and maintained subpopulations more stably than two classical niching techniques (*fitness sharing* and *deterministic crowding*), as well as a classical *EMAS*. Tendency to maintain high diversity within species was observed for *FS*, but in this case there were also agents located outside the basins of attraction of local minima. *DC* has the strong tendency to lose located basins of attraction of local minima during the evolution process. *EMAS* cannot be applied to multi-modal function optimization without introducing special mechanisms such as co-evolution. It turned out that competition for limited resources and environment with a defined spatial structure are not enough to form and maintain more than one species in the case of multi-modal problems.

5 Concluding Remarks

Most of classical niching techniques *indirectly* model resource sharing within the niches. On the other hand, parallel evolutionary algorithms model speciation caused by geographical isolation of subpopulations. The approach of co-evolutionary multi-agent systems combines all these techniques in one coherent model. It allows for natural implementation of the process of *sympatric speciation* based on niche separation due to resource competition. At the same time *allopatric speciation* is achieved based on environmental structure of *EMAS*.

The presented results of simulation experiments are promising and encourage further work on the idea of co-evolution in *EMAS*. It demands more experimental studies to find out which parameters of the introduced techniques are of vast importance for their efficiency, and how the behaviour of the system changes for different kinds of problems.

References

1. E. Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*, 10(2):141–171, 1998.
2. R. Dreżewski. A co-evolutionary multi-agent system for multi-modal function optimization. In M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. J. Dongarra, editors, *Proc. of the 4th Intl. Conf. on Computational Science (ICCS 2004)*, LNCS, Springer-Verlag, 2004.
3. S. Gavrilets. Models of speciation: what have we learned in 40 years? *Evolution*, 57(10):2197–2215, 2003.
4. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.
5. M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In W. I. Grosky and F. Plasil, editors, *SofSem 2002: Theory and Practice of Informatics*, LNCS. Springer-Verlag, 2002.
6. S. W. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature — PPSN-II*, Elsevier, 1992.
7. J. Paredis. Coevolutionary computation. *Artificial Life*, 2(4):355–375, 1995.
8. M. A. Potter and K. A. De Jong. Cooperative coevolution: An architecture for evolving co-adapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

Automatic Transformation from Geospatial Conceptual Workflow to Executable Workflow Using GRASS GIS Command Line Modules in Kepler^{*}

Jianting Zhang, Deana D. Pennington, and William K. Michener

LTER Network Office, the University of New Mexico,
MSC 03 2020, 1 University of New Mexico
Albuquerque, NM, 87131, USA
{jzhang, dpennington, wmichene}@lternet.edu

Abstract. Many geospatial models are developed using command line modules of GIS packages. To utilize scientific workflow technology in geospatial modeling, it is important to support command line GIS modules in scientific workflow systems. However, straightforward representation of command line modules as workflow components conflicts with conventional conceptual design patterns. We propose a two-step geospatial scientific workflow composition approach. Simple conceptual workflows are composed in the first step. These allow data type-based workflow validation. The validated conceptual workflows are then transformed automatically into executable workflows using command line modules in the second step. We describe the preliminary implementation of the proposed approach in the Kepler scientific workflow system and demonstrate its feasibility using an example.

1 Introduction

Traditional geospatial data processing functions are provided in the form of command line modules. They take a set of control options, a set of input/output file names, transform input files into output files and optionally output resulting messages to standard output devices (e.g. screen). The Geographic Resources Analysis Support System (GRASS) is the most widely used open source Geographical Information System (GIS) package. Originally developed by U.S. Army Construction Engineering Research Laboratories in 1982, GRASS has evolved into a large system that consists of more than two hundred command line modules ranging from 2D vector/raster data analysis to 3D visualization and image processing [1].

Scientific Workflow technologies have attracted considerable research and application interests during the past years under the framework of grid computing. In a scientific workflow, data are passed as tokens through component ports, which must be wired for the specific data type expected. Ports can also be used to pass additional specifications required for a particular processing unit, as is the case with components based on command line modules. Configuration of the additional ports is based on the

^{*} This work is supported in part by DARPA grant # N00014-03-1-0900 and NSF grant ITR #0225665 SEEK.

syntactic requirements of the command line modules. Therefore, workflow composition from command line GIS modules can be partially automated by making use of the embedded syntax to handle the additional details, freeing the application developer (domain specialist) to focus on the more conceptual aspects of the workflow.

The most straightforward representation of command line modules as workflow components is to treat the control options and the input/output file names as the inputs of a workflow processing unit. The outputs can be text messages and/or an exit code. However, the conventional conceptual design pattern, i.e., inputs-processing-outputs, is greatly encumbered by representing this subsidiary information along with the high-level information during workflow design. Not only do the control options and file names clutter the conceptual workflow, the representation is not intuitive for a domain user. For example, the output data of a conceptual processing unit is treated as an output but the file name associated with the output data in the corresponding executable processing unit is treated as an input. In addition, it is very difficult to perform semantic validation on workflows that use command line modules based on the compatibility of the input/output data types of the workflow components, because all the input file names are string data type and all the exit codes are integer data type.

We propose a two-step approach to composing geospatial processing workflows: the first step composes a conceptual workflow and the conceptual workflow is transformed automatically into an executable workflow using the embedded information in the command line modules in the second step. The conceptual workflow is closer to users' perception of geospatial processing, which uses semantic geospatial data types (such as vector/raster/tin). The processing units in the conceptual workflows are abstractions of command line modules. Semantic validations can be performed on the conceptual workflows. Only the validated conceptual workflows are allowed to be transformed to executable workflows. The processing units in the executable workflows correspond to the command line modules at the syntactic level. When the executable workflow is executed, the command line options and input/output file names are obtained automatically from the workflow processing units. They are fed to GRASS GIS command line modules that can be either invoked directly within the processing units or output scripts for future execution. While the implementations of the proposed approach are specific for the Kepler scientific workflow system ([2]), we believe the approach is applicable to other scientific workflow systems as well. The proposed approach is motivated by the work in [3] and specifically aims at using command line modules in scientific workflow systems.

The rest of the paper is arranged as follows. Section 2 introduces the Kepler scientific workflow system. Section 3 describes the conceptual and executable workflow component libraries in Kepler. Section 4 presents the proposed automatic transformation approach and provides technical details of the implementations. Section 5 is the demonstration and finally section 6 is the summary and future work directions.

2 Kepler Scientific Workflow System

Kepler [2] builds upon the mature, dataflow-oriented Ptolemy II system (Ptolemy [4]). Ptolemy controls the execution of a workflow via so-called directors that represent models of computation. Individual workflow steps are implemented as reusable

actors that can represent data sources, sinks, data transformers, analytical steps, or arbitrary computational steps. An actor can have multiple input and output ports, through which streams of data tokens flow. Additionally, actors may have parameters to define specific behavior. An illustration is shown in Fig. 1. Note that Parameter Port is an extension of regular IO Port and its value can either be preset using an associated parameter or updated by the connecting port dynamically as a regular IO Port. Kepler inherits and extends these advanced features from Ptolemy and adds several new features for scientific workflows, such as ontology-based data and actor searching, semantic type checking and advanced object management for distributed execution of workflows.

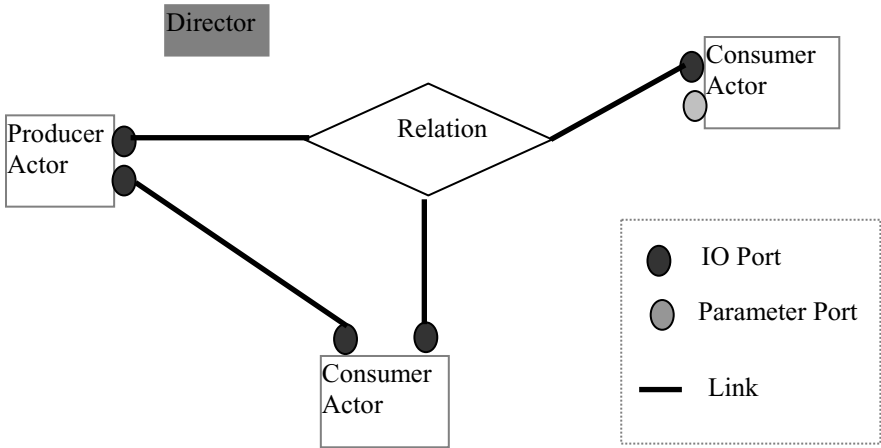


Fig. 1. Illustration of Basic Components in Kepler Scientific Workflow System

Kepler provides an annotated library of workflow components (directors, actors, parameters, etc.). When users drag and drop them into the workflow composition canvas, the data associated with the workflow components are added to the workflow model. Actors and the ports associated with the actors are rendered graphically. Users can then connect two ports or a port and a relation (or creating a *link* in Ptolemy terminology) by dragging and dropping as well. The workflow composition canvas allows typical types of zooming (in/out/fit) and automatic layout. The automatic layout in conjunction with manual adjustments can produce nice visualizations of workflows. In addition, Kepler provides a panner (or a miniature map) to navigate users through complex workflows in a convenient manner.

Workflow components in Kepler/Ptolemy can be specified in XML using Ptolemy’s Model Markup Language (MoML, [4]). Ports and parameters of actors can be added dynamically by modifying the actors’ MoML elements. This feature supports create actors to represent command line modules in a GIS package using a single placeholder actor without actual programming. The composed workflows are internally represented as MoML documents as well. Fig. 2 shows a fraction of MoML document that adds a parameter (property) called “*comments*” and two IO ports (*input_dt* and *output_dt*). It uses a placeholder actor called “*util.ConceptActor*” to

instantiate a geospatial processing actor called “*r_buffer*” that corresponds to “*r.buffer*” module in GRASS GIS package. Note that the “.” symbol in GRASS module naming convention is replaced with “_” in MoML because it is reversed for denoting naming hierarchy in Ptolemy.

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE entity PUBLIC "-//UC Berkeley//DTD MoML 1//EN"
"http://ptolemy.eecs.berkeley.edu/xml/dtd/MoML_1.dtd">
<group>
  <entity name="r_buffer" class="util.ConceptActor">
    <property name="comments" class="ptolemy.data.expr.StringParameter" value="Creates
a raster map layer showing buffer zones surrounding cells that contain non-NULL category
values.">
      <property name="_style" class="ptolemy.actor.gui.style.NotEditableLineStyle"/>
    </property>
    <port name="input_dt" class="ptolemy.actor.TypedIOPort">
      <property name="input"/>
      <property name="type" class="ptolemy.actor.TypeAttribute" value="string"/>
    </port>
    <port name="output_dt" class="ptolemy.actor.TypedIOPort">
      <property name="output"/>
      <property name="type" class="ptolemy.actor.TypeAttribute" value="string"/>
    </port>
  </entity>
</group>

```

Fig. 2. MoML representation of conceptual actor corresponding to *r.buffer*

3 Preparing Actor Libraries

We have developed two separate workflow component libraries. The conceptual workflow component library includes actors necessary for conceptual workflow composition termed as “conceptual actors”. They are jointly developed by Kepler workflow experts and GRASS GIS experts. Basically these conceptual actors are the abstractions of GRASS GIS modules focusing on input and output data types. Controls and optional parameters associated with the GRASS GIS modules are ignored in the conceptual actors for simplicity. The executable workflow component library includes actors syntactically corresponding to GRASS GIS modules. Similarly they are termed as “executable actors”. The graphical representations of the executable actor corresponds to the *r_buffer* conceptual actor in Fig. 2 are shown in Fig. 3. To correctly capture control options and parameters of a GRASS GIS module that an executable actor represents, the actor class of the executable actors (“*util.ExecActor*”) does much more work than “*util.ConceptActor*” which is basically a placeholder.

When an instance of *ExecActor* is executed (or “*fired*” in Ptolemy terminology), the name of the actor is obtained as the initial value of the command line string. Second, the actor checks all its parameters. If the names of the parameters begin with “-“, then they are treated as control options. If the value of such a parameter is “*true*” then its name (such as “-a”) is appended to the command line string. Third, the actor checks the input ports of the executable actor. If the width of the port is zero, which

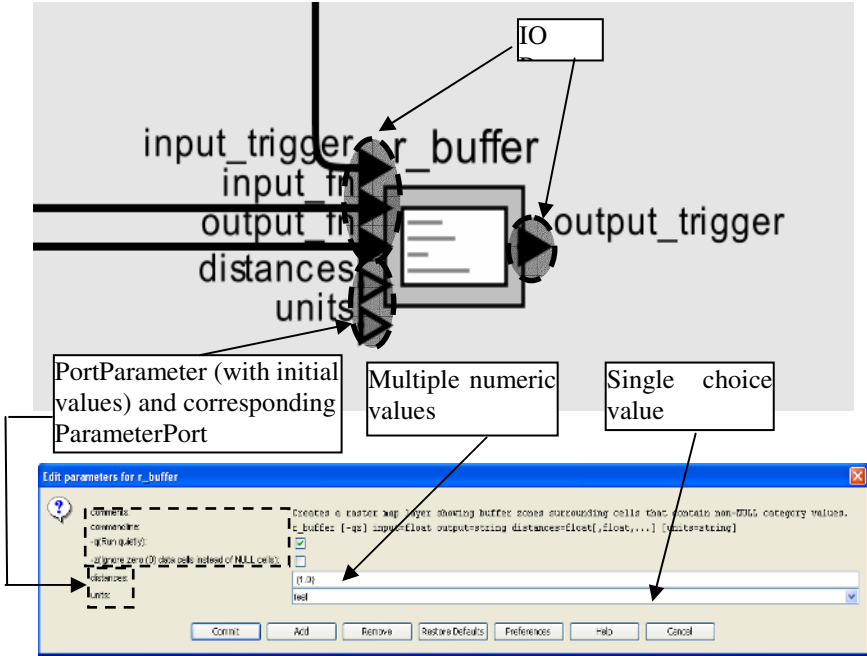


Fig. 3. Graphical representations executable actor *r_buffer*

means no data is fed to the port (if the port is a *ParameterPort* rather than a regular port), the persistent value of the port is obtained. Otherwise the value is obtained from a regular port dynamically. For the port values end with “_fn” (indicating a file name) or embraced by “{” and “}” pair (indicating there are multiple values in between), they are trimmed from the port values. The port names and the trimmed port values form “key=value” pairs and will be appended to the command line string. Once building the command line string is finished, a true value is sent to the actor’s output indicator port (*output_trigger*) to tell the workflow execution scheduler to execute next actor. The command line string can either be used to invoke the GRASS GIS module within the actor execution process or output as part of the script file to be executed in a local or remote computation grid.

4 Automatic Transformation

Supported by Kepler scientific workflow system infrastructure and a few transformation rules (embedded in the text below), we now present the algorithm to automatically transform conceptual workflows to executable workflows.

1. For each conceptual actor in a conceptual workflow, find the corresponding executable actor by looking up its name in the executable actor library.
2. For each connecting conceptual actor pair, find the port named “*output_trigger*” from the executable actor corresponding to the source conceptual actor and the

- port named “*input_trigger*” from the executable actor corresponding to the destination conceptual actor. Connect the two ports in the executable workflow.
3. For each output port of the conceptual actors, find its connecting input port. If the names of the two connecting ports both ends with “*_dt*” (indicating data type), find the ports with the same roots but ending with “*_fn*” (indicating file name) in the corresponding executable actors, create a *Const* actor (representing the file name) and connect the output port of the *Const* actor to the input ports of the two executable actors in the executable workflow with a relation.
 4. For each input port of the conceptual actors, if the port name ends with “*_dt*” and there is a port in the corresponding executable actor whose name has the same root but ends with “*_fn*”, create a *Const* actor and connect the output port of the *Const* actor to the input port of the executable actor in the executable workflow with a relation.
 5. Layout the executable workflow.
 - a. Executable actors are put at the same locations in the executable workflow as their corresponding conceptual actors in the conceptual workflow. The extent of the executable workflow on the composition canvas can be calculated.
 - b. Put all the added *Const* actors at the left side of the executable workflow. All their x coordinates are assigned a fix number (e.g. 50). The y coordinate of the i^{th} *Const* actor can be computed as $i*h/n$ where h is the height of the previously calculated workflow extent at y direction and n is the number of added *Const* actors.
 - c. For the added *Const* actors whose output ports connect more than one port, vertexes are added to the relation associated with the output ports. The added vertexes give more freedom to connect ports with straight lines and produce nicer workflow graphic representations (c.f. Fig. 4 in Section 5).

While the automatic transformation algorithm reduces most of the needed efforts of executable workflow composition (adding actors representing file names, change input/output data ports of conceptual actors to input file name ports of executable actors and connect them correctly), manual post-processing may be required for some complex conceptual workflows. We are in the process of developing more sophisticated algorithms to handle complex conceptual workflows.

5 Demonstration

The demonstrative example is a simple workflow that creates a convex hull from a point data set using GRASS GIS’s *v.hull* module. The convex hull is then rasterized using the *v.to.raster* module. Finally the *r.buffer* module is used to create buffers around the convex hull. The conceptual workflow (Fig. 4 top) is fairly simple and each of the three actors has one input port and one output port. The abstract nature of the conceptual workflow is ideal for validation. We are working towards data types based semantic workflow validation based on some previous work ([5][6]).

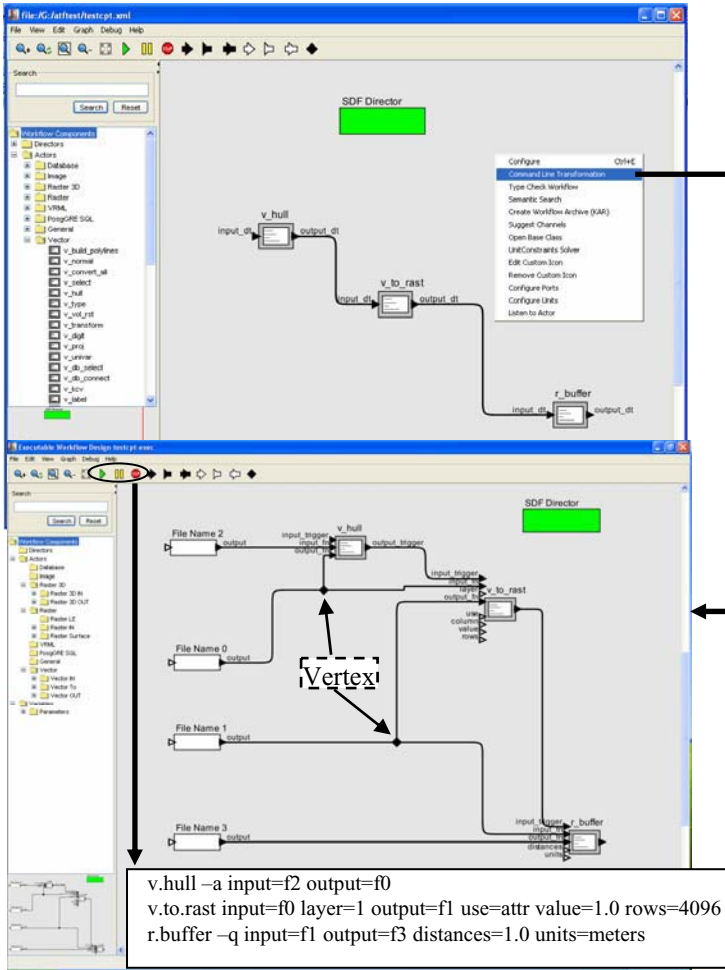


Fig. 4. Conceptual (top) and executable (bottom) workflows of the example

The derived executable workflow (Fig. 4 bottom) is more complex than the conceptual workflow. However, the parameters and the ports of the executable workflow are syntactically compatible with the parameters of GRASS GIS modules. When the executable workflow is executed, GRASS GIS command line scripts can be generated from the executable workflow as shown in the lower part of Fig. 4.

6 Summary and Future Work Directions

We proposed a two-step approach to the composition of geospatial scientific workflows using open source GRASS GIS command line modules. Built on top of Kepler scientific workflow system infrastructure, we developed both conceptual and

executable actor libraries for GRASS GIS command line modules. We also provided a practical algorithm to automatically transform conceptual workflows to executable workflows after users construct and validate the conceptual workflows. A demonstrative example was presented to show the feasibility of the proposed approach and the functionality of the preliminary implementations within Kepler.

For future work, we would like to develop more sophisticated automatic transformation algorithms to handle complex conceptual workflows. While Kepler supports semantic type checking for conceptual workflow design, integration is left for future work. Furthermore, although the implementations currently support GRASS GIS modules only, including modules from other GIS package (such as ESRI ArcGIS) is planned.

References

1. GRASS Development Team: GRASS 6.0 Users Manual. ITC-irst, Trento, Italy. Electronic document: http://grass.itc.it/grass60/manuals/html_grass60/ (2005)
2. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., Mock, S.: Kepler: An Extensible System for Design and Execution of Scientific Workflows. In: Proceedings of the 16th International Scientific and Statistical Database Management Conference (SSDBM) (2004) 423-424
3. Ludäscher, B., Altintas, I., Gupta, A.: Compiling Abstract Scientific Workflows into Web Service Workflows. In: Proceeding of the 15th International Scientific and Statistical Database Management Conference (SSDBM) (2003) 251-254
4. Brooks, C., Lee, E. A., Liu, X., Neuendorffer, S., Zhao, Y., Zheng, H. (eds.): Heterogeneous Concurrent Modeling and Design in Java. Electronic document <http://ptolemy.eecs.berkeley.edu/ptolemyII/> (2005)
5. Bowers, S., Ludäscher, B., Actor-Oriented Design of Scientific Workflows. In: Proceedings of the 24th International Conference on Conceptual Modeling (ER) (2005) 369-384
6. Zhang, J., Pennington, D., Michener, W.K.: Validating Compositions of Geospatial Processing Web Services in a Scientific Workflow Environment, In: Proceedings of the 3rd IEEE International Conference on Web Services (ICWS) (2005) 821-822

A Three Tier Architecture for LiDAR Interpolation and Analysis

Efrat Jaeger-Frank¹, Christopher J. Crosby², Ashraf Memon¹,
Viswanath Nandigam¹, J. Ramon Arrowsmith², Jeffery Conner²,
Ilkay Altintas¹, and Chaitan Baru¹

¹ San Diego Supercomputer Center, University of California, San Diego,
9500 Gilman Drive, La Jolla, CA 92093, USA

{efrat, amemon, viswanat, altintas, baru}@sdsc.edu

² Department of Geological Sciences, Arizona State University,
Tempe, AZ 85281, USA

{chris.crosby, ramon.arrowsmith, jsconner}@asu.edu

Abstract. Emerging Grid technologies enable solving scientific problems that involve large datasets and complex analyses. Coordinating distributed Grid resources and computational processes requires adaptable interfaces and tools that provide a modularized and configurable environment for accessing Grid clusters and executing high performance computational tasks. In addition, it is beneficial to make these tools available to the community in a unified framework through a shared *cyberinfrastructure*, or a portal, so scientists can focus on their scientific work and not be concerned with the implementation of the underlying infrastructure. In this paper we describe a scientific workflow approach to coordinate various resources as data analysis pipelines. We present a three tier architecture for LiDAR interpolation and analysis, a high performance processing of point intensive datasets, utilizing a portal, a scientific workflow engine and Grid technologies. Our proposed solution is available through the GEON portal and, though focused on LiDAR processing, is applicable to other domains as well.

1 Introduction

With improvements in data acquisition technologies comes an increase in the volume of scientific data. The demand for efficient processing and management of the data have made Grid infrastructures an essential component in a wide range of scientific domains. Grid infrastructure technologies enable large scale resource sharing and data management, collaborative and distributed applications and high performance computing, for solving large scale computational and data intensive problems. However, the distributed and heterogeneous nature of Grid clusters, such as various hardware platforms and software systems, access and interaction interfaces and data and resource management systems, make the Grid environment difficult to use by the layman, and thus require additional management to coordinate the multiple resources. In this paper we propose a

coordination of Grid resources in a workflow environment as part of a three tier architecture. The workflow system provides a modularized and configurable environment. It gives the freedom to easily plug-in any process or data resource, to utilize existing sub-workflows within the analysis, and easily extend or modify the analysis using a drag-and-drop functionality through a graphical user interface.

The Geosciences Network (GEON) [1] is an NSF-funded large Information Technology Research (ITR) project to facilitate collaborative, inter-disciplinary science efforts in the earth sciences. GEON is developing an infrastructure that supports advanced semantic-based discovery and integration of data and tools via portals (the GEON portal), to provide unified and authenticated access to a wide range of resources. These resources allow geoscientists to conduct comprehensive analyses using emerging web and Grid-base technologies in order to facilitate the next generation of science and education. One of the challenging problems GEON is currently focusing on is distribution, interpolation and analysis of LiDAR (Light Distance And Ranging) [2] point cloud datasets. The high point density of LiDAR datasets pushes the computational limits of typical data distribution and processing systems and makes grid interpolation difficult for most geoscience users who lack computing and software resources necessary to handle these massive data volumes. The geoinformatics approach to LiDAR data processing requires access to distributed heterogeneous resources for data partitioning, analyzing and visualizing all through a single interactive environment. We present a three tier architecture that utilizes the GEON portal as a front end user interface, the Kepler [3] workflow system as a comprehensive environment for coordinating distributed resources using emerging Grid technologies, and the Grid infrastructure, to provide efficient and reliable LiDAR data analysis. To the best of our knowledge, there exists no previous work on utilizing a scientific workflow engine as a middleware behind a portal environment for coordinating distributed Grid resources.

The rest of this paper is organized as follows. Section 2 provides an introduction to LiDAR data and describes the traditional processing approach. Section 3 gives a brief overview of the Kepler scientific workflow system. The novel approach for LiDAR processing, utilizing the Kepler workflow engine through the GEON portal is described and analyzed in Section 4. We conclude in Section 5 and discuss expansions of the current work aimed at making GEON a leading portal for LiDAR processing.

2 Introduction to LiDAR and Previous Approach

LiDAR (Light Distance And Ranging, a.k.a. ALSM (Airborne Laser Swath Mapping)) data is quickly becoming one of the most exciting new tools in the Geosciences for studying the earth's surface. Airborne LiDAR systems are composed of three separate technologies: a laser scanner, an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS) all configured together to calculate the absolute location for the earth's surface based upon each individual

laser return. The systems typically record one or more returns per square meter on the ground with an absolute vertical accuracy of better than 15 cm. Capable of generating digital elevation models (DEMs) more than an order of magnitude more accurate than those currently available, LiDAR data offers geologists the opportunity to study the processes the shape the earth's surface at resolutions not previously possible. LiDAR data is currently being utilized by earth scientists for a wide variety of tasks, ranging from evaluating flooding hazards to studying earthquake faults such as the San Andreas [2].

Unfortunately, access to these massive volumes of data generated by LiDAR is currently difficult, and the average geoscience user is faced with the daunting task of wading through hundreds or thousands of ASCII flat files to find the subset of data of interest. The distribution, interpolation and analysis of large LiDAR datasets, currently performed on a desktop PC with software packages available and familiar to most earth scientists, also presents a significant challenge for processing these types of data volumes. In the current state of affairs, the popularity and rate of acquisition of LiDAR data far outpaces the resources available for researchers who wish to work with these data. The geoinformatics approach to LiDAR processing described herein represents a significant improvement in the way that geoscientists access, interpolate and analyze LiDAR data. The improved, internet-based approach, acts to democratize access to these exciting but computationally challenging data.

3 Kepler: A Scientific Workflow System

Kepler [4, 5] is a cross-project, multi-disciplinary collaboration to build open source tools for scientific workflows that provide domain scientists with an easy-to-use, yet powerful system for capturing and automating their ad-hoc process. Kepler is built on top of the PtolemyII system developed at UC Berkeley, which provides a set of java APIs for modeling heterogeneous, concurrent and hierarchical components by means of various models of computations [6, 7]. Kepler provides the scientists with a repetitive and configurable environment available through a graphical user interface and as a command-line tool. It combines high-level workflow design with execution and runtime interaction, access to local and remote data and legacy applications, and local and remote service invocation along with a built-in concurrency control and job scheduling mechanism.

Computational units in Kepler are called *actors*, which are reusable components communicating with each other via input and output ports. The control of flow of actors is orchestrated by a *director* that specifies the model of computation. Kepler uses the Modeling Markup Language (MoML), inherited from the underlying PtolemyII system, as its workflow description language. MoML is a modularized and extensible XML modeling language where actors can be defined as place holder stubs to be set prior to the workflow execution. In the next section we describe how utilizing the Kepler features enhances LiDAR processing.

4 A Scientific Workflow Based Approach for LiDAR Processing

In the following section we present a three tiered Kepler scientific workflow solution for facilitating distribution, interpolation and analysis of LiDAR datasets.

4.1 Coordinating Distributed Resources in a Single Environment

LiDAR processing requires three main computational steps each deployed on a distributed resource: querying point cloud datasets, processing the data using various interpolation algorithms, and visualizing the results. Coordinating these steps in a Kepler scientific workflow provides scheduling and monitoring of each task and communication between the various resources. Furthermore, the scientific workflow environment gives us modularity and extensibility through reusable actors. The LiDAR processing workflow, depicted in Figure 1, provides a *conceptual* workflow where each of the components can be dynamically customized by the availability of the data and processing algorithms and is set on the fly prior to the workflow execution. This modularized approach can be captured as a workflow pattern of *subset*, *analyze*, and *visualize*. Below we elaborate on each of the processing steps.

The massive amount of LiDAR data (currently two datasets at ~1 billion points each) were uploaded and organized in a DB2 spatial database on *DataStar* to provide a unified structure to the collected data. The subset query returns all points (X,Y,Z) that reside within a user selected bounding box. The database connection information along with the query are specified as workflow parameters and are set on the fly prior to the workflow execution. The query is then performed on DataStar where the result is also stored on an NFS mounted disk.

The analysis step consists of an interpolation algorithm. As shown in the figure, the query response is shipped to the *analysis* cluster and is then interpolated

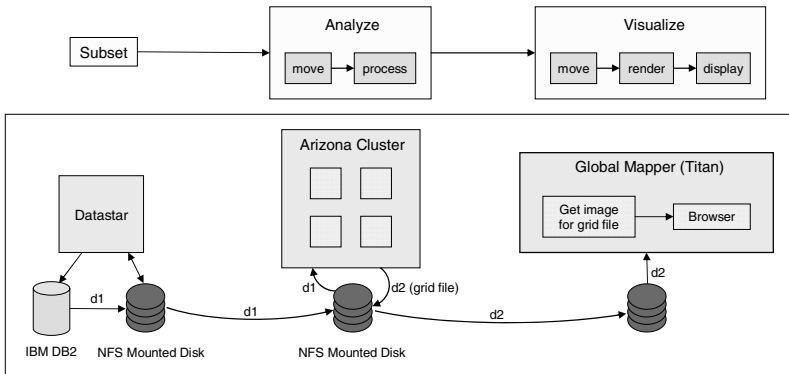


Fig. 1. Coordinating distributed heterogeneous resources in a single scientific workflow environment

into a regularized grid. Currently we use the GRASS spline interpolation [8] algorithm deployed on the Arizona GEON four nodes cluster. Other interpolation algorithms, for example, Inverse Distance Weighted (IDW) or Kriging algorithms may be plugged in as well. Interpolation of the high-point density LiDAR data currently constitutes the bottleneck in the overall process. Parallelization of the interpolation code or alternative interpolation algorithms will likely result in improved performance.

The interpolation results may be visualized and/or downloaded. At present, the Global Mapper imaging tool¹ is used to create a display from the interpolated results. Global Mapper, available to the GEON community through a web service, takes an ASCII grid file as an input and returns a URL to the resulting image. The image can be displayed on a web browser and requires no specific visualization components. Other visualization methods may be applied as well.

4.2 A Three Tier Architecture

The GEON project is developing a web-based portal for gathering GeoScience applications and resources under a single roof. In order to make the workflow based approach for LiDAR processing uniformly accessible through the GEON portal, our proposed solution is based on a three tier architecture (depicted in Figure 2): the *portal layer*, the *workflow layer* or control layer, which is the Kepler workflow system, and the *Grid layer* as the computation layer.

The *portal layer*, a portlet, serves as a front end user interface. It enables the user to partition the data using an interactive mapping tool and attribute selection through a WMS map². Algorithms, processing attributes and desired derivative products, are also chosen using a web interface. Within Kepler, one can design a predefined parameterized workflow template which is modularized and configurable using place holder stubs to be set prior to the workflow execution. The aforementioned “*subset, analyze, visualize*” workflow pattern serves as a conceptual workflow template, defined in the Kepler workflow description language, MoML. A workflow instance is created on the fly from the conceptual workflow based on the user selections. The instantiated workflow is then scheduled to be executed by the workflow layer.

The *workflow layer*, also referred to as the main control layer, communicates both with the portal and the Grid layers. This layer, controlled by the Kepler workflow manager, coordinates the multiple distributed Grid components in a single environment as a data analysis pipeline. It submits and monitors jobs onto the Grid, and handles third party transfer of derived intermediate products among consecutive compute clusters, as defined by the workflow description. In addition, it sends control information (a.k.a. tokens) to the portal client about the overall execution of the process. The workflow is executed by the Kepler engine in a batch mode. Once a job is submitted, the user can detach from the system and receive an email notification after the process has completed.

¹ <http://www.globalmapper.com/>

² OpenGIS Web Mapping Specification (<http://gisserver.esrin.esa.int/quickwms/>).

As the LiDAR processing workflow involves long running processes on distributed computational resources under diverse controlling authorities, it is exposed to a high risk of component failures, and requires close monitoring. In order to overcome these failures with minimal user involvement, Kepler provides a data provenance and failure recovery capability by using a job database and smart reruns. The job database is used for logging the workflow execution trace and storing intermediate results along with the associated processes/components that were used to produce them. The workflow engine maintains information about the status of each intermediate step, and this can be used to initiate a smart re-run from a failure point or a checkpoint. These advanced features thus eliminate the need to re-execute computationally intensive processes.

The *Grid layer*, or the execution layer is where the actual processing implementations are deployed on the distributed computational Grids. Currently a simple submission and queueing algorithm is used for mapping jobs between various resources based on the number of allocated tasks and the size of the data to be processed. In the near future we plan to utilize the Pegasus [9] Grid scheduler to benefit from mapping jobs based on resource efficiency and load, thus making the process more robust. We also plan to extend this further by deployment of the computationally challenging processes on a higher performance machine, for example, DataStar, which consists of 32 P690 processors with 128GB of memory running at 1.7GHz, each with a gigabit connection to an underlying SAN disk infrastructure, making it an ideal machine for compute intensive tasks.

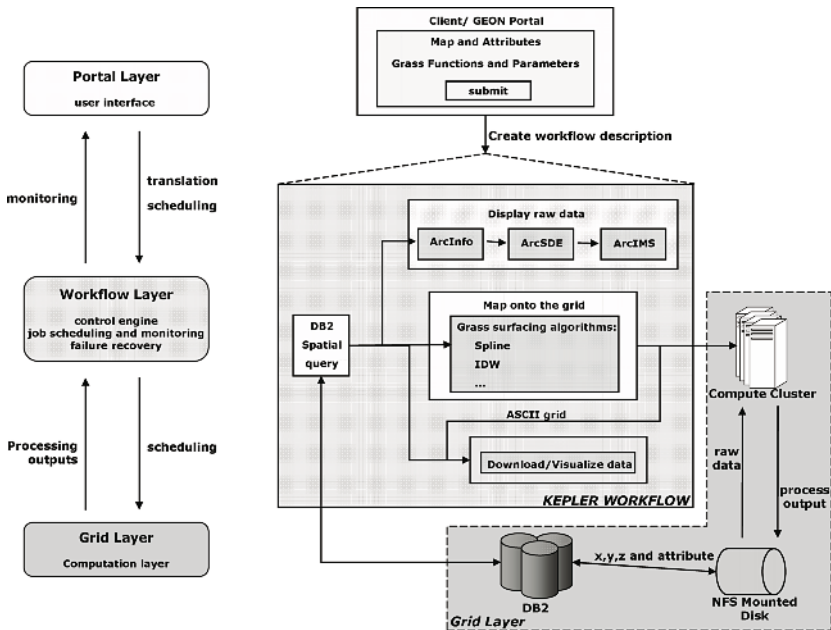


Fig. 2. Three tier LiDAR processing architecture

4.3 Data Upload and Access

As initial pilot datasets, LiDAR data collected along the Northern San Andreas Fault (NSAF) in Sonoma and Mendocino Counties, California and data from the Western Rainier Seismic Zone in Pierce County, Washington were utilized. The approximately 1.2 billion points in each of these datasets were stored in a DB2 Spatial Extender Relational Database on DataStar. To optimize query performance, and due to its sheer volume, the data were grouped by USGS Quarter Quadrants³ and organized in the database as a table per quarter quadrant. The multiple data segments are accessible through the WMS map using a metadata table that specifies the bounding box coordinates for each segment. Each record in the datasets consists of a coordinate, elevation and corresponding attributes. The Spatial Extender feature in DB2, leveraging the power of standard SQL for spatial data analysis, enables storage, access and management of the spatial data. Furthermore, it provides a grid indexing technology for indexing multi-dimensional spatial data, dividing a region into logical square grids, thus facilitating region based subset queries.

4.4 Analysis of the Approach

The three tier based architecture utilizes the Kepler workflow system to reengineer LiDAR data processing from a Geoinformatics approach. The data and processing tools are accessible through a shared infrastructure and are coordinated using the Kepler engine. Kepler provides an incremental design and development environment which allows users to create incomplete workflow templates, which can be filled on demand prior to the workflow execution. Seamless access to resources and services is enabled through existing generic components such as SSH, Web Service, database access and command line processor actors. These generic building block components offer the flexibility to plug-in any applicable data or process, thus providing a customizable and modularized environment. The LiDAR workflow currently has access to two datasets both stored in a DB2 database on DataStar. Other databases sharing a similar structure may be used simply by pointing to a different database location. Currently we use GRASS spline interpolation, available via a web service deployed on the Arizona GEON cluster. Other interpolation algorithms such as GRASS IDW can be applied by updating a WSDL URL parameter. The visualization tools are interchangeable as well. Global Mapper can be replaced with FlederMaus⁴ or ArcIMS visualization tools, both deployed on the GEON portal as web services.

The LiDAR workflow's main processing bottleneck is in the interpolation of the high point density datasets. Currently, the GRASS spline interpolation is limited to processing 1,600,000 points. This problem is being addressed by the GRASS development community and we anticipate a solution to this problem in the near future. Ultimately however, to improve performance, a parallel interpolation algorithm is required along with deployment on a higher performance

³ <http://www.kitsapgov.com/gis/metadata/support/qqcode.htm>

⁴ An interactive 3D visualization system (<http://www.ivs3d.com/products/fledermaus/>).

machine. We are also testing the Pegasus system [9] in coordination with the Pegasus group at the University of Southern California for a more efficient mapping of the interpolation sub-workflow onto the Grid.

5 Conclusion

In this paper we describe a three tier architecture for LiDAR data processing using a comprehensive workflow system, a shared cyberinfrastructure and evolving Grid technologies. The first version of this effort is available at the GEON portal (<https://portal.geongrid.org:8443/gridsphere/gridsphere>), and has already been incorporated as a public tool. We plan to extend this work in progress by making additional datasets available, and improving the overall performance with advanced processing tools such as parallel interpolation algorithms and enhanced visualization methods. We also intend to utilize the Kepler provenance system to link the workflow execution trace to the portal interface in order to provide extended user monitoring of the workflow execution status. Our goal is to provide a centralized location for LiDAR data access and interpolation that will be useful to a wide range of earth science users.

Acknowledgement. GEON is supported by NSF ITR 0225673. Special thanks to Dogan Seber and Ghulam Memon for their support in this work.

References

1. NSF/ITR: GEON: A Research Project to Create Cyberinfrastructure for the Geosciences, www.geongrid.org
2. Carter, W.E., Shrestha, R.L., Tuell, G., Bloomquist, D., and Sartori, M., 2001, Airborne Laser Swath Mapping Shines New Light on Earth's Topography: Eos (Transactions, American Geophysical Union), v. 82. p. 549
3. Kepler: An Extensible System for Scientific Workflows, <http://kepler.ecoinformatics.org>
4. Ludäscher B., Altintas I., Berkley C., Higgins D., Jäger-Frank E., Jones M., Lee E.A., Tao J., Zhao Y.: Scientific Workflow Management and the Kepler System. Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, to appear, 2005
5. Altintas I., Berkley C., Jäger E., Jones M., Ludäscher B., Mock S.: Kepler: Towards a Grid-Enabled System for Scientific Workflows, in the Workflow in Grid Systems Workshop in The Tenth Global Grid Forum, Germany, 2004
6. Lee E. A. et al.: PtolemyII Project and System. Department of EECS, UC Berkeley, <http://ptolemy.eecs.berkeley.edu/ptolemyII>
7. Liu X, Liu J., Eker, J., Lee E. A.: Heterogeneous Modeling and Design of Control Systems, in Software-Enabled Control: Information Technology for Dynamical System, Tariq Samad and Gary Balas, Wiley-IEEE Press, 2003
8. Mitsova, H., Mitsova, L. and Harmon, R.S., 2005, Simultaneous Spline Interpolation and Topographic Analysis for LiDAR Elevation Data: Methods for Open Source GIS, IEEE GRSL 2(4), pp. 375- 379
9. Blythe J., Jain S., Deelman E., Gil Y., Vahi K., Mandal A., Kennedy K.: Task Scheduling Strategies for Workflow-based Applications in Grids. CCGrid 2005

Workflows for Wind Tunnel Grid Applications

A. Paventhan¹, Kenji Takeda¹, Simon J. Cox¹, and Denis A. Nicole²

¹ Microsoft Institute for High Performance Computing,
School of Engineering Sciences,
University of Southampton, UK

{pavs, ktakeda, sjc}@soton.ac.uk
² School of Electronics and Computer Science,
University of Southampton, UK
dan@ecs.soton.ac.uk

Abstract. Aerodynamicists use wind tunnels to aid the research, design and development of products such as aircraft, cars, and yachts, amongst others. The data acquired from such tests must be acquired, collated, processed and analysed in a timely fashion to maximise productivity. In many scenarios a variety of data acquisition systems are used, and managing the overall testing process can be challenging. The wind tunnel grid project aims to provide an extensible, network-based system that can provide a more seamless working environment for scientists and engineers, so that they can focus on the data analysis and interpretation part of the process.

In this paper we describe the development and implementation of the wind tunnel grid system workflow. By exploiting Windows Workflow Foundation we are able to provide an easy-to-use and extensible workflow environment (wind tunnel grid workflow framework) that meets the requirements of both the developer and end-user well. By leveraging the .NET-based CoG Toolkit previously developed, interoperability with Globus grid services is demonstrated.

1 Introduction

The importance of workflows while developing real-world scientific and grid applications is becoming increasingly apparent [1]. While many business workflow systems tend to be control flow driven, many scientific workflows are data-centric.

Current Grid computing [2] solutions allow large-scale multi-institutional access to distributed resources. There are many issues, however, still to address while implementing application-specific scientific workflow on grids.

We can categorize scientific workflow systems as below: 1) Workflow systems motivated by application driven system requirements. 2) Generic workflow systems. The extensibility of a domain-specific workflow solutions to another application domain is limited as it does not address other sets of requirements. On the other hand, a generic workflow solution would require further customization to be applicable to a particular application domain. The set of requirements for scientific workflow systems in different scientific disciplines, in terms of data

size, formats, realtime requirements and computational complexities, bring different sets of challenges. Considering the Wind Tunnel user requirements (see Section 3), a customized workflow is important due to the varied nature of the work, which changes on a job-by-job basis in a multi-user facility .

In this paper we present an approach to building customized scientific workflows based on WinFX Windows Workflow Foundation. Windows Workflow Foundation is an extensible framework for developing workflow solutions with predefined set of activities (IfElse, While, Parallel, InvokeWebService and so on) and support for building domain-specific custom activities by inheritance. By making use our earlier work, the multi-language Commodity Grid Kit [3], we deliver a wind tunnel grid workflow framework for application development.

The rest of the paper is organized as follows. In Section 2 we discuss related works. Section 3 outlines wind tunnel experiment requirements to illustrate why workflow customization is important. Section 4 covers a brief overview of Windows Workflow Foundation. In Section 5, we present our approaches to wind tunnel grid workflow, with conclusions and future work presented in Section 6.

2 Related Works

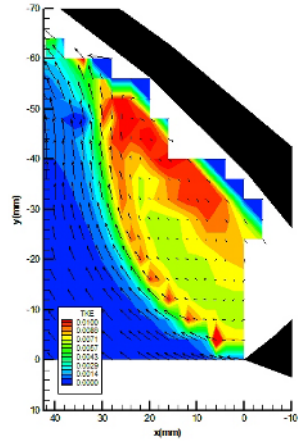
In this section, we discuss the influences of related works on our project and highlight the particular wind tunnel experimental requirements that lead us to our work.

The GriPhyN [4] project addresses the workflow requirements of physics experiments. On data request, an abstract workflow is constructed. It is then converted to concrete workflow represented as Condor DAGman files [5] for submission to Condor-G scheduler. In case of wind tunnel experiments, the workflow is triggered by data acquisition and the raw data transfer requires more customization. In the KEPLER system [6], the workflow components are known as *actors* and controlled by an object known as a *director*. KEPLER addresses Grid and web services access, but, the integration of data acquisition hardware and experiment specific data transfer are paramount in our work. Grid-DB [7] is a data-centric grid workflow system. It provides a declarative language to define workflow. DiscoveryNet [8] addresses the need for knowledge discovery process in lifesciences. The components and workflows in DiscoveryNet are composed as Web and Grid services by sharing across teams. While the Triana [9] problem solving environment demonstrates distributed workflow to implement gravitational wave search algorithm.

The wind tunnel experiments run on several proprietary systems and their integration into scientific workflow requires customized solutions. Also, the data transfer and processing requirements are experiment specific, with the issues of physical locality before, during and after any given test a significant issue . The wind tunnel workflow framework is aimed at providing user with ready-to-use experiment specific workflow activities, hiding the underlying complexities and at the same time providing user with an option for customization, if required. Over time the processes and systems available in the wind tunnel change significantly, so the ability to rapidly develop and customise workflows is a necessary requirement.



(a) Aircraft landing-gear test in wind tunnel using microphone arrays



(b) Typical LDA visualization [11]

Fig. 1. Wind Tunnel Experiments

3 Wind Tunnel Experiments

The wind tunnel facilities at University of Southampton [10] are typical in that they house a variety of specialized experimental hardware and software for academic and industrial research. They are used in a wide variety of projects including fundamental aerodynamics, aerodynamics of racing cars and road vehicles, rotorcraft aerodynamics, aeroacoustics, aeronautics, wind engineering and industrial aerodynamics. Currently, in many wind tunnel experiments, the data movement operations to the processing computer are manual due to interoperability issues between hardware, software and the acquisition systems. The automated solution would require the following steps: 1. Experiment-specific data verification to ensure whether the acquisition was indeed successful 2. Experiment-specific annotation of metadata for auto-upload and processing 3. Raw data movement operations (based on metadata) and 4. User-defined processing step. We discuss below the requirements of a typical wind tunnel experiment that has been used to demonstrate the Grid system.

3.1 Laser Doppler Anemometry (LDA)

LDA systems use non-intrusive point-measurement techniques to accurately measure fluid velocity in highly turbulent or reversing flow. For each experimental configuration, a calibration step must be performed and a transformation matrix must be derived. LDA data acquisition software collects selected number of samples (typically thousands) at user programmed traverse positions of up to three velocity components (This value is also equal to number of Burst Spectrum Analysers (BSA)). The collected data are stored in separate raw data files for each traverse position. The raw data filenames have a suffix 0, 1 or 2 to indicate the velocity component (u, v & w , in the laser coordinate system). The file extension

represents the traverse position. There are essentially $n \times p$ raw data files for one experiment, n ($n=3$) is the number of velocity component and p is the number of traverse positions. The user parameters for acquisition are stored in a separate flat file. The upload activity of the workflow requires verification of the raw data files prior to the uploading to processing node. Since the number of velocity components and traverse positions are known from the parameter file, all the raw data files along with metadata (transformation matrix, user parameters) can be uploaded without any user intervention. The LDA processing steps include data conversion, coincidence processing, moment processing, spectrum processing and correlation processing. The user should be able to run the default algorithm or provide the new one by overriding the default.

The other experiments that we work on with similar requirements include: 1. The microphone phased array technique used in aeroacoustic research with near-realtime upload and processing requirements. and 2. Particle Image Velocimetry (PIV) with binary image upload and high-performance cross-correlation processing requirements. As can be seen from the requirements, the metadata must be used for customized data upload. In addition, the user processing step also requires customization so that different algorithms can be developed and used as the state-of-the-art advances.

4 Windows Workflow Foundation

Microsoft Windows Workflow Foundation is an extensible framework and is part of the upcoming Microsoft's next generation development Framework, WinFX [12]. The workflow in Windows Workflow Foundation is composed from a set of *activities*, compiled to a .NET assembly and is executed on the workflow runtime and the Common Language Runtime (CLR).

4.1 Workflow Model and Composition

There are two models supported [13]: 1. Sequential Workflow Model—comprising activities that execute in a predictable sequential path, and 2. State Machine Model—a flow driven by events triggering state transitions. In both these models the basic element of the workflow is called an *activity*. Some of the Windows Workflow Foundation's activity types include: control-flow (While, IfElse, Delay), exception (throw, exception-handler and BPEL compensations), data handling (Update, Select), transactions (and compensations for long-lived "transactions" that cannot be directly unwound) and Communication (InvokeWebService, InvokeMethod). A workflow consists of metadata for the workflow definition and the accompanying .NET classes that form the code file. The workflow can be composed using a visual workflow designer, which has a drag-and-drop interface or declaratively in XOML, an XML dialect for writing workflows. The workflow can also be completely coded in CLR languages. A workflow must be compiled with *wfc* workflow compiler before it can be run. All the Windows Workflow Foundation *activities* are derived from *Activity* base class. The Windows Workflow Foundation extensible development model enables creation of

domain-specific *activities* which can then be used to compose workflows that are useful and understandable by domain scientists.

4.2 Workflow Runtime, Scheduling and Hosting

The Workflow runtime layer is at the core of Windows Workflow Foundation and is responsible for execution, tracking, state management, scheduling and policies. The workflow engine runs inside a *hosting process* provided by the workflow application. The hosting layer is responsible for communication, persistence, tracking, transaction, timing and threading. It is possible to dynamically update the running workflows on the fly. With this flexible approach to workflow hosting and extensible framework for workflow activities, most of the functionality of the state-of-the-art scientific workflow systems [14] can be hosted on top of Windows Workflow Foundation. In the following sections we illustrate how specific workflows to wind tunnel aerodynamic testing can be constructed using Windows Workflow Foundation.

5 Wind Tunnel Grid Workflow

In this section, we describe architecture of the wind tunnel application workflow and show how customized solutions can benefit the wind tunnel users.

5.1 Application Workflow Leveraging Windows Workflow Foundation

Our approach to implementing a wind tunnel grid workflow based on Windows Workflow Foundation is shown in Figure.2. Users are able to use local services, Web Services and Globus Grid services using our previously developed MyCoG.NET commodity grid toolkit [3]. Three different sets of activities are available to compose an experimental workflow: 1. Windows Workflow Foundation Activities 2. MyCoG.NET-based Grid activities to access Globus services 3. Experiment-specific activities for upload, processing, results and so on. The user can design the workflow from these activity sets depending on his or her requirements.

There are two possible options to host the workflow: 1. Client-controlled hosting, and 2. By submitting to the wind tunnel grid workflow server for hosting. In client controlled hosting, the workflow runtime runs as part of the *host process* running on the user's PC. In this case, the user has to leave the *host process* running until the workflow finishes. Underlying the hosting process is the WinFX workflow runtime and .NET Common Language Runtime. The workflow can be monitored from wind tunnel grid workflow client while it is running. In the second case, the user deploys their workflow for hosting to the wind tunnel grid workflow server after successful Grid Security Infrastructure (GSI) [15] authentication and delegation of user credentials. The wind tunnel grid workflow server maintains user account information. A separate *host process* is instantiated for the user's workflow and the runtime is started. This allows the user to disconnect after submission and monitor the workflow periodically from wind tunnel grid workflow client.

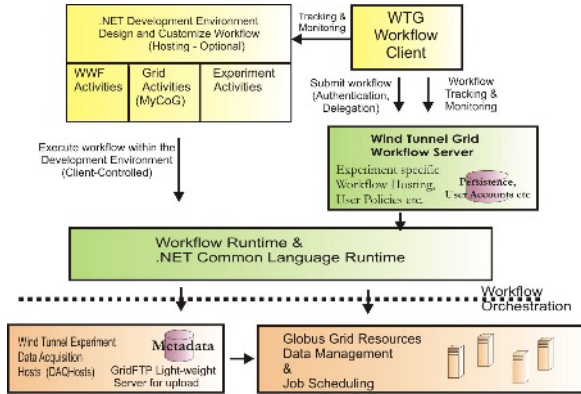


Fig. 2. Wind Tunnel Experimental Workflow Architecture

The generic wind tunnel workflow activities are: *WTWInit* - initializes wind tunnel workflow server hosting process for the user; this is the first activity in any wind tunnel experimental workflow. *UserNotification* - Customized user notification on the state of the workflow (workflow completion or failure). Figure.3 shows a sequential LDA workflow designed using customized wind tunnel grid workflow activities. The *WaitForDAQ* is an event driven activity customized for the LDA experiment. On completion of the data acquisition, this activity verifies the raw data files for completeness and workflow transitions to next activity. The *MyGridFTP*, *MyGram* and *MyMDS* activities use the *MyCoG.NET* Commodity Toolkit to access Globus resources. The implementation details of *MyCoG.NET* are discussed in [3]. These Grid service access activities are further customized for individual experiments. For example, as shown in Figure.3, the *LDAUpload* activity derived from *MyGridFTP* has specific input properties for the experiment (data acquisition hostname, number of data points, number of burst spectrum analysers). Some properties are initialized at workflow design time with default values and others received as input from the *host process*. The experiment specific properties would enable, for example, automatic uploading of raw data files from data acquisition host to a Gram server, as can be seen from Section.3.1. The *FetchResults* is an activity derived from *MyGridFTP* to transfer results from Gram host to Windows Workflow server and to the user's desktop.

The microphone array processing algorithm is significantly more processor intensive than the LDA case. A similar workflow can be constructed using Globus services, or Windows-based HPC resources. This is currently carried out using Labview and Matlab, but a C# version of the processing algorithms is being developed and integrated using Windows Workflow Foundation.

We specifically highlight the ease with which the Windows Workflow version of the LDA system was constructed. Due to the ease-of-use and intuitive nature of the workflow GUI, the user is able to drag-and-drop activities onto the work surface, rapidly customising their own workflow without worrying about lower-level details.

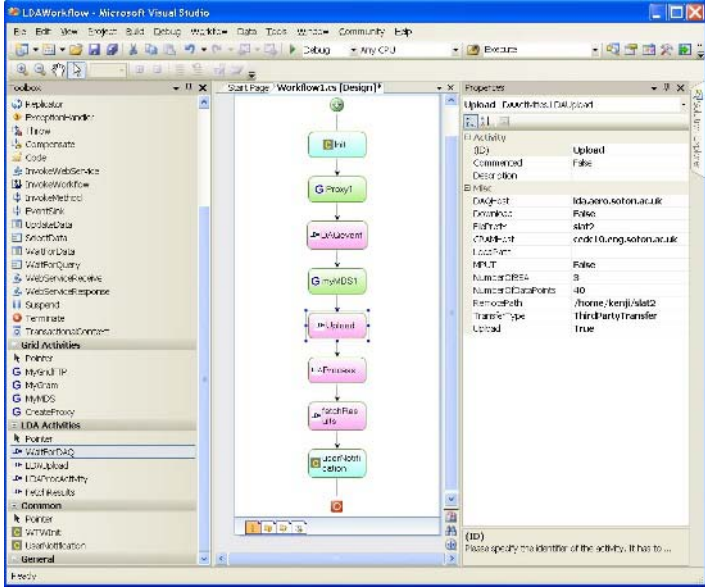


Fig. 3. LDA workflow

In our earlier implementation [3] of LDA workflow, each stage of the workflow is triggered by the user via portal interface. By bringing Globus grid service access to the Windows Workflow Foundation using MyCoG, the user would be able to design, execute and monitor wind tunnel experimental workflow on Globus resources.

This is of significant benefit to the wind tunnel engineer, who previously had to deal with the workflow steps on different hardware using bespoke software tools. The new system provides benefits not only in terms of speed, but reliability and consistency for the testing process and subsequent analysis.

6 Conclusions and Future Work

In this paper we have discussed the implementation of real-world scientific workflows using Windows Workflow Foundation in wind tunnel applications. This provides an easy-to-use, customisable and extensible framework to create sequential or state-based workflows. We have shown, using the MyCoG.NET commodity grid toolkit, that interoperability between Windows Workflow Foundation and Globus grid services is possible, including certificate-based authentication, proxy support, GridFTP file transfer and GRAM job submission. In this way Windows Workflow Foundation can be used to orchestrate workflows between Microsoft Windows-based Services, other Web Services and Globus Grid services.

While the paper describes an application-specific example of using Windows Workflow Foundation, it is designed as a generic framework. It is therefore suitable for other branches of experimental and computational sciences as well. Due

to its extensibility, the addition of provenance tracking, semantic definition and representation, and metadata derivation from results is possible.

As more grid resources move to Web Service interfaces, the use of Windows Workflow Foundation as a generic framework for composing scientific workflows is likely to become more common.

References

1. Ludascher, B., Goble, C.: Guest Editors' Introduction to the Special Section on Scientific Workflows. *ACM SIGMOD Record* **34**(3) (2005)
2. Foster, I., Kesselman, C.: *The GRID 2: Blueprint for a New Computing Infrastructure*. second edn. Morgan-Kaufmann (2003)
3. Paventhan, A., Takeda, K.: MyCoG.NET: Towards a multi-language CoG Toolkit. In *3rd ACM International Workshop on Middleware for Grid Computing*. (2005)
4. Deelman, E., Blythe, J., Gil, Y., Kesselman, C.: "Workflow Management in GridPhyN" in *Grid Resource Management J. Nabrzyski, et.al, eds.* Kluwer (2003)
5. Condor DAGman. (<http://www.cs.wisc.edu/condor/dagman/>)
6. Ludscher, B., Altintas, I., et.al: Scientific workflow management and the KEPLER system. (*Concurrency and Computation: Practice & Experience*, (to appear))
7. Liu, D.T.: The design of GridDB: A Data-Centric Overlay for the Scientific Grid. In: *Proceedings of the International Conference on Very Large Data Bases*. (2004)
8. Curcin, V., Ghanem, M., et.al: IT Service Infrastructure for Integrative Systems Biology. In: *IEEE International Conference on Services Computing*. (2004)
9. Churches, D., Gombas, G., et.al: Programming scientific and distributed workflow with triana services. *Concurrency and Computation: Practice & Experience*, (to appear) (2005)
10. Southampton wind tunnels. (<http://www.windtunnel.soton.ac.uk/>)
11. Takeda, K., Ashcroft, G.B., Zhang, X., Nelson, P.A.: Unsteady aerodynamics of slat cove flow in a high-lift device configuration. *AIAA Paper 2001-0706* (2001)
12. WinFX Developer Center. (<http://msdn.microsoft.com/winfx/>)
13. Andrew, P., Conard, J., et.al: *Presenting Windows Workflow Foundation*, Beta Edition. Sams (2005)
14. Yu, J., Buyya, R.: A taxonomy of scientific workflow systems for grid computing. *ACM SIGMOD Record* **34**(3) (2005) 44–49
15. Welch, V.: *Grid Security Infrastructure Message Specification*. (2004)

Distributed Execution of Workflows

Ismael Navas-Delgado¹, Jose F. Aldana-Montes¹, and Oswaldo Trelles²

¹ University of Malaga, Computing Languages and Computer Science Department,
29071, Malaga, Spain
{ismael, jfam}@lcc.uma.es
<http://khaos.uma.es>

² University of Malaga, Computer Architectures Department,
29071, Malaga, Spain
ots@ac.uma.es
<http://www.ac.uma.es>

Abstract. In this work we present a tool for manually connecting services by means of their input and output data, helping bioinformaticians in the automatic evaluation of workflows for the dynamic integration of heterogeneous data sources, services and computational resources. Our workflow platform offers a view of the different tools available as a single and uniform pool of services readily available for enhancing query processing.

1 Introduction

A Web-based service facilitates access to remote resources promoting the development and availability of highly diverse and specific tools. These new resource capabilities are of special interest in the bioinformatics domain where a mixture of databases and services are required to produce a more complete view of biological problems. Unfortunately, the common bioinformatics research field is hard to operate in since it can involve the localization of adequate web services by collecting URLs of the useful ones, selecting the most popular or qualified ones, getting familiar with their specific interfaces (e.g. see the very popular sites like NCBI, EBI, ExPASy, etc.) copying and pasting data, manually selecting and combining partial results, and building by-hand and scheduling workflows. Thus, the main component of a bioinformatician's daily work consists in carrying out a set of simple activities that they usually perform using a diverse set of tools for solving a problem. This implies interacting with different interfaces and storing partial results, in order to use them subsequently in another tool. This manual interaction with services is costly and prone to error.

The use of a workflow-based system is very useful when the tasks that the user needs to solve are usually predefined and the relationships between the tasks are well known. Our proposal provides an execution environment for related tasks, so that users can develop a workflow (making use of an external graphical tool) defining a set of related tasks for solving a specific problem, and can then execute the workflow with specific inputs. This process can be repeated several times with different inputs in order to derive biological conclusions.

In this paper we focus on the description of a workflow management environment that provides a set of applications for storing, executing and monitoring workflows defined by means of XML files in the Scuf1 [1] representation language.

2 System and Methods

The process of defining and executing workflows that we propose is composed of several steps (see Figure 1): (1) To define workflows manually (writing an XML Scuf1 file) or using a graphical tool for this language, such as Taverna Workbench, which allows users to graphically describe the services to be executed and how they are related; (2) To store user or generic workflows in the platform making use of the provided web interface; (3) To specialise workflows by defining their inputs, specific values for simple data objects or links to existent complex BioMOBY objects; (4) To execute workflows, taking the inputs and then executing services as soon as their inputs are available; (5) To monitor the execution, noting the changes in the status of the processors and analysing the obtained data.

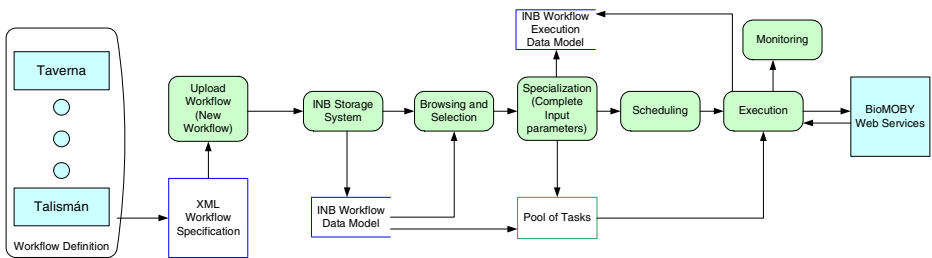


Fig. 1. Data and Process Flow for using the workflow management platform. Blue nodes are external applications or services, and green ones are internal processes.

Internal Representation and Storage

Our platform offers an interface for loading, executing and monitoring the execution of workflows (see Section 2.3). This platform offers a persistent database to store workflows in order to improve their use and their analysis (see Figure 2). This database contains a set of tables that can be queried to retrieve workflow information or for executing the workflow (Workflow, Processor, Link, Source and Sink). In the current version we offer the capability of executing BioMOBY services and scripts related with them, like those for getting the inputs and showing the outputs (*Create_moby_data* and *Parse_moby_data*).

Each processor is connected to a workflow (by means of the Id), and those processors are related by means of links, in which one processor acts as a source and the other as a sink. Then, sources of a workflow are linked by means of the link table with a processor (source data can be used in several processors). In the same way, the sinks of a workflow are linked to the output of a processor (usually each sink is related to only one processor).

The database tables designed include all the elements for defining a workflow (composed of BioMOBY services), so that there is no loss of information in the

storage process. Besides, the use of constraints allows verifying the correct insertion in the database, and additionally it is possible to rebuild a workflow (in XML format) from the database. Thus, if a workflow is published and shared between different users, all of them can retrieve the XML description of the workflow in order to make use of it.

When a workflow is loaded, it is parsed and its elements (processors, links, sinks and sources) are stored in the database and a copy of the XML document (describing the workflow with Scuffl) is uploaded onto the INB web server. Besides, additional information is added to workflows in order to improve the documentation of loaded workflows: Name, Short Description and an optional Long Description (as documentation of the workflow). This information is essential if we want to share our workflows with other people, because (only) the title of a workflow (information included inside the XML description) is usually insufficient.

Executing and Monitoring Workflows

The platform provides a web interface for searching workflows, which shows a short description of each workflow (so users have an initial description that could be useful for selecting a service). Once a workflow is selected from the list, the next step is to define the inputs of the workflow in order to execute it. When a user inserts or selects the input/s, the system stores information about the workflow that is going to be executed. Thus the inputs are stored for execution and future use. In addition, a set of internal tasks are created (one for each processor), and related processors imply that the corresponding tasks have as input the output of the predecessor. Relationships between the designed tables are quite similar to the relationships between the tables for storing the workflows, though the sources include a field for storing the value inserted by the user.

The created tasks (that are related to the processors by means of the `IdTask` field) are executed by the system scheduler, which does not execute a task until the required input has been created (the objects have a state that indicates whether an object has been created or not). Thus, the synchronization between processors is ensured.

The workflows, which a user has executed, can be monitored by means of a web interface that shows the execution of the workflow with automatic refreshing of the information of the web page. This monitoring tool also includes all the created tasks, their state: `Waiting`, `Finished` or `Error`, and their outputs. Thus, the partial and final results can be examined in order to extract biological conclusions to the executed workflows.

3 Conclusions

The work described in this paper has been developed and implemented in the National Institute for Bioinformatics (INB) in Spain, and the INB platform is available at <http://www.inab.org/MOWServ>. The INB system architecture is organized at three main levels, with the user minimally armed with a Web-browser, and demanding services to process their collection of biological data: (a) a web-interface at the top of the architecture facilitates communication between the user and the platform (b) the architecture core including services' interface through bioMOBY API; and (c) at the

bottom of the scheme the services' providers. A web interface manages user sessions with an authentication mechanism. An automatic web interface builder is able to dynamically build on interfaces for browsing data objects, services and namespaces (associated with data containers).

At internal level, once a service has been requested, the system provides notification about the progress status of services, -including historical records of executed tasks-, together with the relationships between input data, applied services and output data. Frequently output data becomes the input for new services. The GUI provides a specific list of suitable services that can be applied.

In summary we report a client engine, which is based on semantic interconnection concepts. The platform is able to integrate various processing services developed by different users and groups into workflows, through a web-based interface, thereby expanding the functionality of current services and enabling the easy incorporation of new procedures to customize the system for specific concerns.

The support for loading, executing and monitoring workflows is based on a very common and well-defined representation language. Our proposal extends this tool with the capability of using authentication-based systems, in which the confidentiality of the data is ensured. In addition the use of a scheduler which has statistics about the services that are stored in the system enables a better execution process. Another important advantage of this system is that data obtained by means of the execution of a workflow can be used to execute other services or even workflows.

Acknowledgements

This work has been partially supported by grant "GNV5-Bioinformática Integrada" from Genoma-España and the Spanish MEC Grant TIN 2005-09098-C05-01. We would also like to thank all the other nodes in the INB infrastructure for their heavy involvement in the development of services and integration procedures.

Reference

1. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M. Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P. (2004) Taverna: A tool for the composition and enactment of bioinformatics workflows *Bioinformatics Journal* 20(17) pp 3045-3054, 2004

Applying Workflow to Experiment Control in Virtual Laboratory

Łukasz Czekierda and Krzysztof Zieliński

Distributed Systems Research Group
Department of Computer Science
AGH – University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
{luke, kz}@ics.agh.edu.pl

Abstract. Virtual Laboratory (VLab) has been developed as a distributed component system supporting a remote access to physical devices via the Internet. Unlike in typical VLabs, much attention has been paid to designing a state-of-the-art architecture which facilitates among others exposing the functionality of the devices, composing them into complex experiment stands as well as supervising in an interactive or automated way. The paper describes an application of workflow to experiment control in the VLab. Nevertheless, the presented concept consisting in wrapping a workflow engine by a component of a selected platform component has much more general nature.

1 Motivation of the Work

Virtual laboratories become popular nowadays. WWW resources direct us usually to simple simulation programs from various disciplines implemented using Java applets, ActiveX controls, etc. There are also environments which offer users an access to real devices; the paper concentrates on VLabs understood according to such a meaning.

Virtualization technique makes it possible to see the equipment of the laboratory not as a collection of heterogeneous physical devices but as software entities that wrap their functionality and provide with a well-defined interface for remote management and configuration. This technique is widely accepted and does not pose a considerable problem (using such technologies as Web Services), thus access via Internet to devices such as radio-telescopes is currently almost an everyday practice.

Current virtual laboratories are to a great degree static. Devices are often bound with their graphical user interface and sometimes with each other. Users are not able to change the administratively pre-set order of actions and customize the equipment to their needs. Nevertheless, there are a few exceptions to this, e.g. [2, 3].

A next generation VLab should offer much more features. The most crucial requirement is a clear separation of representation of the device functionality (its instrumentation) from usage scenarios. Thanks to the separation, both a single device or set of them consistently interconnected can be controlled in many various ways depending on goals of performed activities, user's permissions etc.

Interface exposed by the instrumented devices is usually tied to GUI used to influence the experiment course. In some cases the same activities could be performed by an

agent – the user representative. Some experiments can be realized in a mixed mode – the routine parts are performed automatically and the user takes a decision in crucial moments. Experiment logic may be programmed by the VLab administrator. However, in most cases it is better to allow users to do it by themselves. Performing the task must be feasible even for people with no much programming skills – thus they must be provided with a proper tool – user-friendly yet powerful. Amongst many techniques available, using workflow seems to be the best solution.

2 Component-Based Architecture of Virtual Laboratory

VLab as most of the contemporary distributed systems has a layered architecture.

The lowest layer of the VLab – physical devices layer – is composed of a collection of heterogeneous physical devices. The second layer – instrumentation layer – provides devices' representatives exposing a generic management and configuration interface. For representation of virtualized devices CORBA Component Model (CCM) [1] components have been chosen, their interface is expressed using Interface Definition Language (IDL). CCM object definition clearly states which channels (interfaces) the object accepts, which interfaces it uses and which events it emits and consumes, what definitely much better characterizes the object interface than the traditional object-oriented languages.

Thanks to the well-known CCM channel interface it can be automatically determined whether particular devices can be linked via given channels or not. When the interfaces are not fully compatible, an adapter can be used. This conglomerate can be called the experiment stand and is presented in Fig. 1.

Some channels are devoted to external control – they can be attached to the user elements or to elements playing the role of the user's agents. The user should be allowed to supervise the operation of the agent. Agents and user application constitute two upper layers of the architecture of the Virtual Laboratory – experiment and presentation, respectively. They have been also implemented as CCM components.

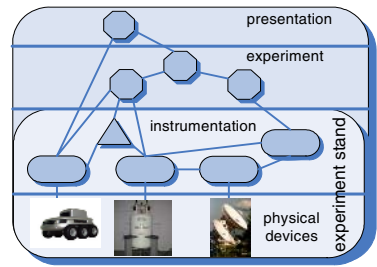


Fig. 1. Multilayer architecture of Virtual Laboratory

3 Embedding Workflow Engine Inside a Component

The idea implemented in the VLab has a much more general nature consisting in encapsulating a workflow engine by a component of a given component platform. This concept is illustrated in Fig. 2. The benefits of such approach are following:

- transparency and uniform representation – the object can be seen and used as any other component (including assembling, configuration and deployment).

- precisely defined interface – workflow environments are able to run any process instance, so the workflow engine lacks a well-defined interface with the external world. Embedding it in the component forces to specify the exact interface. Achieving the goal is in general not a straightforward undertaking. To the most important issues which need to be considered belong:
 - transforming the interface and communication – the concept assumes linking two distinct worlds – external one with RPC-based communication and internal one with other or not specified mechanism. Usually, providing an intermediate layer bridging the communication will be necessary.
 - providing the management interface – besides the interface used for communication with external partners, the component ruled by workflow needs an additional channel – for introducing process definition, starting, stopping or repeating it, etc.
 - monitoring the workflow – workflow engines can be usually monitored what allows users to e.g. check the progress of the workflow execution.
 - evaluating portability and efficiency – workflow engines may need many special libraries and artifacts to run. Moreover, running them with only business process instance requires loading all of the anyway. Thus, lightweight engines are desired.

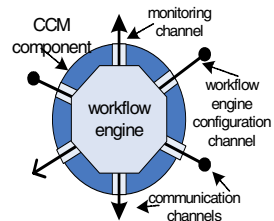


Fig. 2. Placement of workflow engine inside CCM component

In the context of the VLab, the concept has many advantages. All components used in the experiment are uniformly represented and can be assembled together as well as easily scheduled and deployed. With the ability not only to connect required elements but also to define their behavior, users can construct complete experiments.

4 Implementing the Concept in CCM Component Platform

From among a few workflow platforms investigated (BPEL, YAWL), Java Business Process Management (jBPM) [5] has been chosen. It easily integrates with chosen CCM platform implementation – java-based openCCM from ObjectWeb.

jBPM is implemented according to UML activity diagram and is declared to support most of workflow patterns specified by van der Aalst [4]. jBPM combines declarative and procedural approach of process behavior definition – its core is expressed in XML-like jPDL language and actions (handlers) assigned with some events can be written in java. This increases expressiveness of the workflow.

Process definition in jBPM consists of many nodes connected via transitions. Nodes are visited by a token passing from a start to an end node. There are several types of nodes; from the point of view of this discussion two are the most important:

- state nodes – store the token (pointing an active node) until an external event moves it to a next node. State node is then a stable state.
- ordinary nodes – there can be assigned a piece of work to be done (action) to them. The token leaves the node immediately after executing the action.

jBPM does not directly support remote communication (especially CORBA). In case of outgoing invocations (to external CORBA objects) it is possible to implement handlers assigned with the node which are able to perform any action – in particular do remote CORBA invocations. Incoming communication is much more difficult. In authors' implementation incoming operation invocations are parsed, converted by a special intermediate layer to so called signals and assigned with a proper token. As a result of receiving the signal, the token leaves the current state node, passes through several nodes executing defined actions and stops in the next state node.

It is worth to emphasize roles of the VLab administrator and users in this context. Administrator specifies the component's interface and implements the handlers. The single handler should encapsulate a simple functionality so as to be able to build the behavior of the component in a fine-grained way. Constructing the workflow using existing elements can be done by the users provided with a proper (graphical) tool.

5 Applying the Concept in the Virtual Laboratory – Case Study

The presented architecture has been implemented. To the collection of virtualized devices belong mobile robots, a localization system, a video streaming server, and a pan-tilt camera. Several experiments have been designed and implemented using these devices, most of them give full control over the experiment course to the users.

The benefits of using workflow in the VLab are evidently visible especially in case of experiments supervising a technological or computational process where finishing task. Nevertheless, even using the mentioned virtualized devices it is possible to successfully apply workflow to experiment control.

The experiment stand contains the mobile robot and the localization system which observes it. The robot is expected to come from the starting to the end point via several destinations on the experiment area with some obstacles placed. The role of workflow is to lead it so as to avoid obstacles. Nevertheless, hitting one of them triggers notification received by the workflow and forces to return to the previous destination successfully reached. The experiment layer CCM component is connected to: (1) several input channels notifying about reaching a destination, (2) input channel notifying about hitting an obstacle or entering a forbidden area, (3) output management channel of the robot driving its wheels, (4) input management channel coming from the user supervising the experiment and able to start, pause and finish it.

References

1. Object Management Group. CORBA Components, June 2002, formal/02-06-65.
2. Fjeldly T A. Shur M. S., Lab on the Web, Running Real Electronics Experiments via the Internet, Wiley&Sons, IEEE Press, 2003.
3. Lawenda M., Meyer N., Workflow with Dynamic Management Scenarios in the Virtual Laboratory, 6th CARNet Users Conference, ISBN 953-6802-04-X, September 27-29, 2004.
4. van der Aalst W.M.P., ter Hofstede A.H., Kiepuszewski B., Barros A., Workflow Patterns.
5. Koenig J. JBoss jBPM, White Paper, November 2004.

Integration of Compute-Intensive Tasks into Scientific Workflows in BeesyCluster^{*,**}

Paweł Czarnul

Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology, Poland
pczarnul@eti.pg.gda.pl

Abstract. The paper presents design, implementation details and simulations of scientific workflows involving compute-intensive tasks on clusters and PCs. The author has incorporated support for scientific workflows into previously developed J2EE-based BeesyCluster, deployed at Academic Computer Center Gdansk Poland on large HPC resources including a large 288-processor Itanium2 cluster. BeesyCluster allows users to manage various accounts on clusters/PCs via WWW/Web Services, run shell interactively, compile, queue, run tasks, publish services for other users, work in teams. A frequent scenario in HPC computing is analyzed, in which a workflow is combined from tasks offered by different users. Steps of the workflow include data preparation and following simulations run in parallel on clusters, with and without queuing systems.

1 Introduction and Related Work

In today's distributed networks, services are combined into scientific workflows enabling convenient execution of frequent scenarios in various disciplines like molecular biology, bioinformatics, ecology, chemistry, often using HPC resources.

Examples of systems supporting workflows include Kepler ([1]), Pegasus ([2]), Triana ([3]), P-GRADE ([4]). Such systems may utilize Web Services or grids e.g. Kepler ([1]) features Web Service or grid actors – WEBSERVICE and GLOBUSJOB for calling Web Services and running Globus jobs respectively. Directed Acyclic Graph Manager (DAGMan) is a meta-scheduler for Condor jobs (described by a DAG).

Potential obstacles in integration in current grid ([5]) technology and systems may include: often difficult and time-consuming deployment of new resources, quickly changing versions of standards and software, complexity of the middleware for average users.

For integration of Web Services there exist Web Services Flow Language (WSFL), its extension Services Workflow Language (SWFL), BPEL and WS-Choreography.

The J2EE-based BeesyCluster ([6]), implemented by our team, deployed at Academic Computer Center Gdansk, Poland, is an advanced access portal to HPC resources/PCs and services available on them. While the implemented workflow features are similar to other systems, unlike many middlewares BeesyCluster accesses clusters/PCs via SSH. Thus attachment of resources, unlike for many grid systems, can be done in seconds since does not require any additional software layer on clusters/PCs.

* Partially covered by the Polish National Grant KBN No. 4 T11C 005 25.

** Calculations carried out at the Academic Computer Center in Gdansk, Poland.

2 Integration of Services into Workflows in BeesyCluster

As an exemplary workflow in BeesyCluster we consider the one in Figure 1. User `iccs` runs a `DataGenerator` on a Linux workstation `TBC2` (Pentium 4 2GHz) which prepares input data and runs two parallel simulations on two different clusters: `holk` (288 Itanium processors, Infiniband) with PBS and `parowiec` (16 processors, Ethernet) with no queuing system. Applications are offered by different users in BeesyCluster.

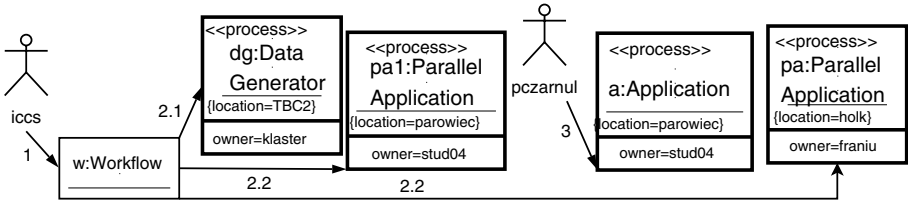


Fig. 1. Basic Interaction Diagram for Testbed Workflow

We have implemented a workflow that responds dynamically to the changes of load on cluster `parowiec` (application `a` started by `pczarnul`), by checkpointing/restarting or migrating `pa1` using our checkpointing libraries for MPI programs ([7]).

In BeesyCluster, services (tasks run interactively, queued or any services that operate on any of resources attached to clusters/PCs) process input data and produce output results. Workflows in BeesyCluster can be modeled as a directed graph with:

- node of a workflow graph** – contains calls to at least one service. A path of connected graph nodes is implemented as a message-driven bean thread on the J2EE platform calling services on clusters/PCs via SSH. The workflow node is represented by a row in a database with service data. The graph links the node with following ones.
- parallel execution of workflow nodes/paths** – threads representing parallel paths are created and execute services in parallel.
- synchronization of parallel workflow nodes on the following node** – threads representing parallel paths can determine through synchronization on the database which finishes last and it continues with services associated with the following node.
- communication between parallel workflow nodes** - inter-thread interaction in the application server or by direct interaction (hand-coded) if nodes on the same resource.
- input/output data management** – output files of preceding service(s) transferred as input files with proper name mapping.

For the testbed scenario (Figure 1) additional services are created (Figures 2, 3):

1. `l1-14` on cluster `parowiec` for load monitoring – implemented as a C++ program `loadmonitor` using command `ps` and parsing output periodically,
2. `apml` on cluster `parowiec` – reads the load provided by `loadmonitors`, checkpoints/stops `pa1` when the processor load by other user processes exceeds 10% and restarts when it drops below or migrates to other nodes immediately; `apml` sends signal `USR1` to checkpoint `pa1` and restarts by a system command ([7]).

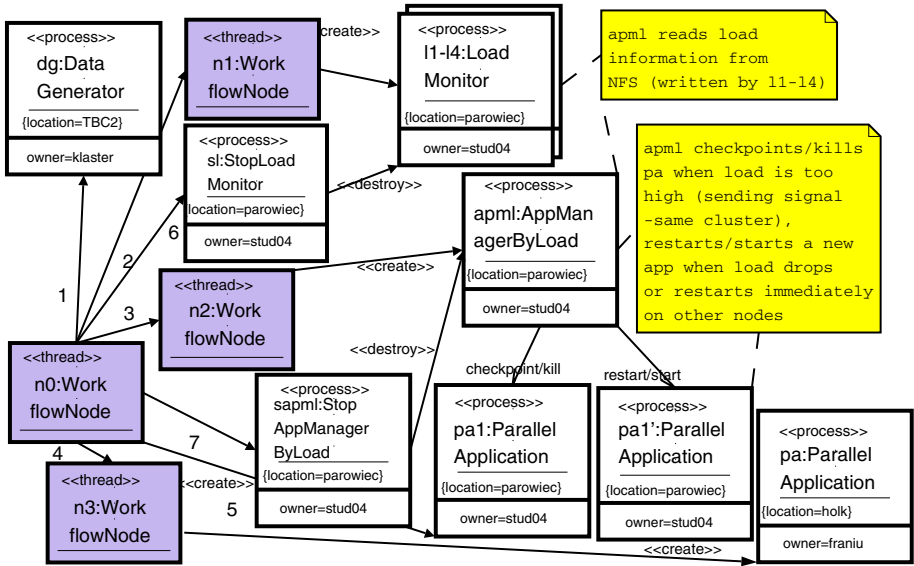


Fig. 2. Cooperation Diagram for Testbed Workflow with Reaction to External Load on Cluster

3 Simulations

Figures 3, 4 and 5 present an interaction diagram and results from the execution of the workflow respectively. In the scenario from Figure 4 pa1 is restarted on the same node when the load from other users drops while for Figure 5 migrated to another node immediately. On holk requesting more nodes requires longer to obtain resources (queued via PBS). On parowiec tasks are run at once (no queuing) but the cluster has

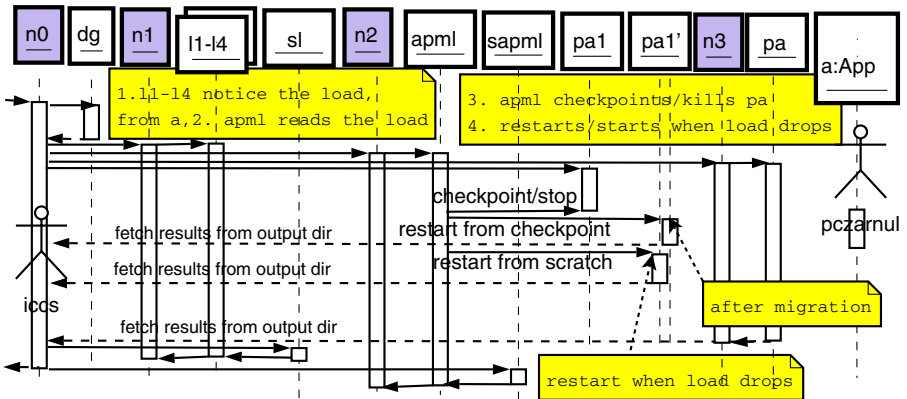


Fig. 3. Interaction Diagram for Testbed Workflow

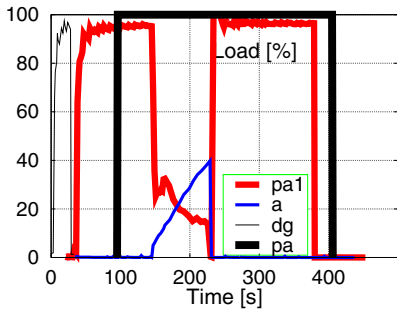


Fig. 4. Workflow with Checkpoint/Restart

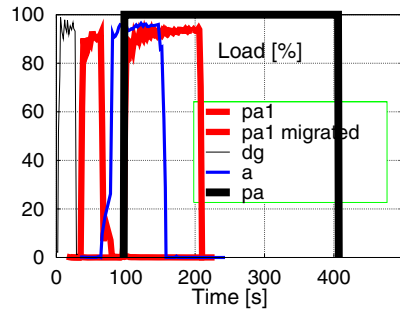


Fig. 5. Workflow with Migration of `pa1`

slower processors and the task can be slowed down by others as in our scenario. For the one in Figure 4 twice as many processors were used on `parowiec` than on `holk`.

4 Conclusions and Future Work

We have presented design and execution of a testbed scientific workflow within BeesyCluster. The workflow responded automatically to launching a higher priority application on cluster `parowiec` by checkpointing/migrating/restarting application. Future work includes incorporation of Web Services and Globus jobs into the system.

References

1. Ludscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows* (2005)
2. Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.H., Vahi, K., Livny, M.: Pegasus : Mapping Scientific Workflows onto the Grid. In: *Across Grids Conference, Nicosia, Cyprus* (2004) <http://pegasus.isi.edu>.
3. Majithia, S., Shields, M.S., Taylor, I.J., Wang, I.: Triana: A Graphical Web Service Composition and Execution Toolkit. In: *IEEE International Conference on Web Services (ICWS'04)*, IEEE Computer Society (2004) 512–524 <http://www.trianacode.org/>.
4. Laboratory of Parallel and Distributed Systems, MTA SZTAKI, Hungary: (Parallel Grid Runtime and Application Development Environment, User's Manual, ver. 8.4.2)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In: *Open Grid Service Infrastructure WG. (2002) Global Grid Forum*.
6. Czarnul, P., Bajor, M., Fraczak, M., Banaszczyk, A., Fiszer, M., Ramczykowska, K.: Remote task submission and publishing in beesycluster : Security and efficiency of web service interface. In Springer-Verlag, ed.: *Proc. of PPAM 2005. Volume LNCS 3911.*, Poland (2005)
7. Czarnul, P., Fraczak, M.: New user-guided and ckpt-based checkpointing libraries for parallel mpi applications. In Springer-Verlag, ed.: *Proceedings of Euro PVM/MPI 2005, 12th European PVM/MPI Users' Group Meeting. Volume LNCS 3666.*, Sorrento, Italy (2005) 351–358

A Distributed Re-configurable Grid Workflow Engine

Jian Cao, Minglu Li, Wei Wei, and Shensheng Zhang

Department of Computer Science & Technology, Shanghai Jiaotong University,
200030, Shanghai, P.R. China

{cao-jian, li-ml, wei-wei, sszhang}@cs.sjtu.edu.cn

Abstract. Grid workflow, as a basic service in the grid environment, is a useful tool to help researchers make use of various grid resources to solve scientific problems. It is possible lots of users invoke grid workflow services in a very narrow time interval. Therefore, a centralized grid workflow engine is apt to be a bottleneck. In the paper, a novel grid workflow engine is proposed. It is based on Jini platform and employs Jini services to implement functions of grid workflow engine. By adding or removing enactment services, the grid workflow engine can be reconfigured dynamically. The workflow manager of the engine can allocate requests to proper enactment services according to some load-balancing strategy. The structure and mechanisms of this grid workflow engine are discussed in the paper. A prototype system and testing results are also introduced.

1 Introduction

In order to solve complex scientific problems, geographically distributed and heterogeneous resources should be connected together through the high-speed network. Grid, which provides virtual organization with capabilities to solve the problems cooperatively, is put forward to meet these more and more critical requirements [1]. As a new infrastructure, grid is extending its role as a high performance computing environment and bringing deep impacts on the human life and the society [2].

To make these complex grid applications accessible to the many potential users outside the scientific community is a great challenge. In earthquake science, for example, integrated earth sciences research for complex probabilistic seismic hazard analysis can have greater impact, especially when it can help mitigate the effects of earthquakes in populated areas. In this case, users might also include safety officials, insurance agents, and civil engineers, who must evaluate the risk of earthquakes of certain magnitude at potential sites [3]. A clear need exists to isolate end users from the complex requirements necessary for setting up earthquake simulations and executing them seamlessly over the grid. Workflow has also been used to describe, control the logic and data flows among grid services so that the burdens of the human to properly set up the grid applications are released. Because of its importance, grid workflow is regarded as a basic service in grid environment.

Since grid is a global resource sharing platform, the amount of users is tremendous. With more and more critical applications running on grid, the grid workflow systems will suffer heavier and heavier load. Such grid applications must execute hundreds or thousands workflow instances everyday. Therefore, the performance of

the grid workflow system becomes one of the key issues to achieve the success of grid applications. The centralized workflow engine in the grid workflow system is very possible to become a source of system failures and a bottle-neck for the system performance since the load may exceed its design capacity.

In the paper, a novel grid workflow engine is proposed. It is based on Jini platform and employs Jini services to provide functions of grid workflow engine. It can be expanded by adding new enactment services and client request can be allocated to a proper enactment service according to a load-balancing strategy.

The remainder of this paper is organized as follows. Section 2 presents a service based grid workflow engine. Section 3 introduces the structure and main function of workflow manager. Section 4 describes a prototype system and some testing results. Section 5 discusses related work. Finally, Section 6 provides some concluding remarks and gives an introduction to some future work.

2 A Service-Based Grid Workflow Engine

Grid workflow engine plays a central role for workflow enactment. We proposed a service-based grid workflow engine, which is divided into two service layers: management layer and enactment layer (See Fig.1). Both of them are composed of a set of services. The enactment layer as a whole provides those functions which are supported by most workflow engines. But all those functions are implemented as services and for each function, there may be several corresponding services, i.e., all these services provide the same function. The management layer is the core layer of distributed engine, which provides access, management and monitor interfaces to the enactment service layer. It is also composed of several services.

As shown in Fig.1, the management layer and enactment layer will communicate with each other and our implementation is based on Jini network. Jini is a new computing paradigm introduced by Sun Microsystems that can provide a network wide

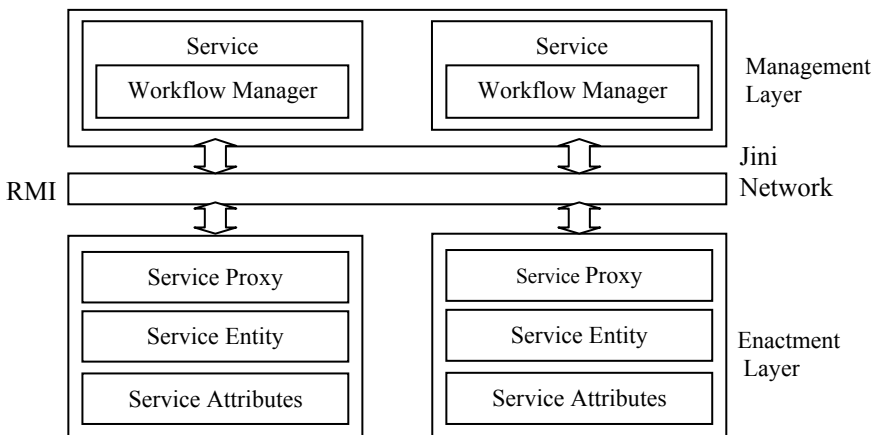


Fig. 1. A Service-based Grid Workflow Engine

plug-and-play environment [4]. The services are the entities that can be used by a person, a program, or another service within the federation. The infrastructure supports the discovery and joint protocol that enables services to discover and register with lookup services. Jini technology provides us with much functionality to establish a flexible, stable and extensible distributed computing environment.

3 The Structure and Main Functions of Workflow Manager

3.1 The Structure of Workflow Manager

In the management layer, each service wraps a workflow manager. The main functions of each workflow manager includes enactment service proxy caching, load balancing for enactment services, remote event response, enactment service management, persistent data storage and client request dispatching. It can be further divided into several parts (See Fig.2):

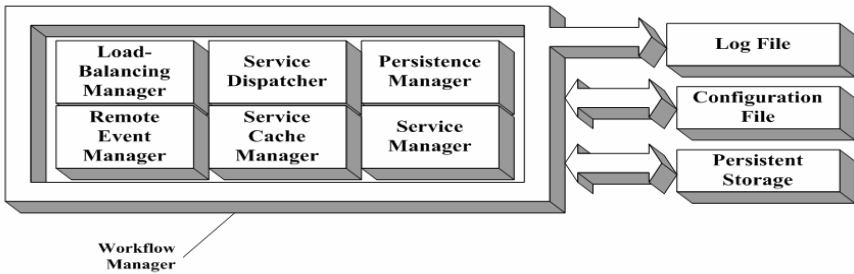


Fig. 2. The Internal Structure of Workflow Manager

- Service Cache Manager (SCM): to maintain an enactment service cache and be responsible for dynamically discovering enactment service. SCM will cache distributed enactment service proxy in local machine and maintain the content of service cache automatically when it finds any status change of available enactment services. For example, SCM may delete unavailable enactment service record or update the attributes of some enactment service so that all enactment services kept in the cache are available and their information is always correct.
- Service Manager (SM): to provide interfaces to manage enactment service remotely. For example, stop or launch a service, modify the attributes of service or service registry information in Jini network.
- Load-Balancing Manager (LBM): to be responsible for managing the load distribution within enactment services. LBM dynamically computes and adjusts the load of each enactment service according to some load-balancing strategy in order to keep the whole distributed environment in a balance status.
- Service Dispatcher (SD): to receive all the user requests, check the user authority and judge the validity of user requests. Only valid user requests could be accepted and processed. According to the workflow execution strategy and global

load-balancing strategy, SD finds a proper enactment service to handler user request and returns the result to user.

- Remote Event Manager (REM): to be responsible for receiving and processing all events happening during the execution of enactment services.
- Persistence Manager (PM): to store the persistent data during the execution of workflow. Persistent data includes service log, configuration file of workflow manager and all information of enactment service proxy in service cache and so on.

Since management layer provides the uniform access interfaces to the client, the process of accessing, managing and configuring enactment services is simplified.

3.2 System Re-configuration

Since this grid workflow engine is based on services, whenever services join in or be removed from the enactment layer the system is reconfigured. The process that enactment services join in workflow manager is shown in Fig. 3. Firstly, enactment service uses Jini Discovery Protocol to find lookup service in Jini network. When lookup service is discovered, it uploads its own proxy and some attributes' information to lookup service through Jini Service Join Protocol. At the same time, workflow manager obtains the enactment service proxy and check whether there is a new enactment service or not. Therefore whenever new enactment services enters into Jini network, the workflow manager will be notified and then download service proxy and store it into the service cache.

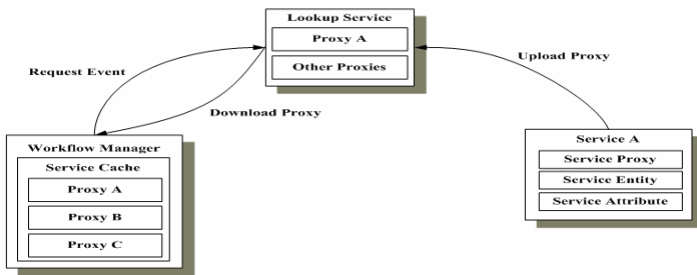


Fig. 3. The Process of Service Joining in Workflow Manager

3.3 Client Requests Handling

The process that workflow manager handles requests from client is shown in Fig 4. When workflow manager receives a request from the client, it passes the requests to the service dispatcher. The service dispatcher determines whether there is a specific service responsible for responding to this type request or not according to a predefined rule. If there is, this related service is invoked. Otherwise, service dispatcher asks load-balancing manager to select a proper service. Load-balancing manager will select a service proxy from the service cache according to a load-balancing strategy and return this service proxy to service dispatcher so that this service can be invoked.

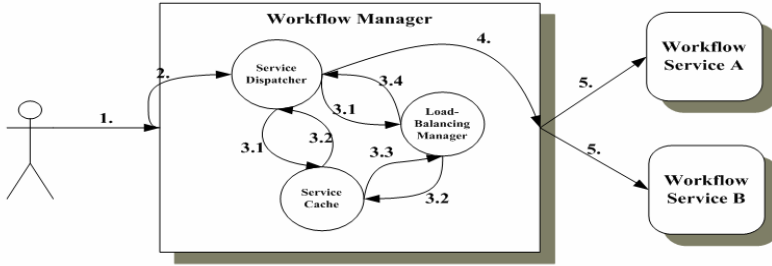


Fig. 4. The Process of Accessing Services through Workflow Manager

3.4 Load Balancing

Currently load-balancing manager uses a simple dynamic load-balancing algorithm to maintain the stability of load distribution in grid workflow system. The load of each enactment service is calculated according to the function:

$$WL = \sum_{i=1}^n P_i W_i$$

where P_i is the value of a enactment service attribute, for example the average response time or the waiting time for handling requests. W_i is the weight of this attribute. Load-balancing manager computes load of each enactment service at regular intervals and selects the service with best performance to handle the request from clients.

4 A Prototype System

We have implemented a prototype system called SGWE (Service-based Grid Workflow Engine), which includes management layer, enactment layer and a management tool. Enactment services mainly include workflow model instantiation, starting a task, submitting a task, accessing work items and renting remote event.

Fig. 5 shows the management tool we developed. In this tool, all available enactment services and executing tasks are listed and monitored. The parameters of each service, such as the number of concurrent users, the max number of concurrent users, CPU usage percent and memory usage percent are also shown. All messages exchanged in different parts of workflow system, such as the execution result of enactment service and the response of remote event are also recorded and shown in this tool.

In order to evaluate the performance of SGWE, experiments were performed to analyze average response time of client request in such a distributed environment. In order to simulate real client access patterns, a request sequence was generated by using a random number generator to produce requests in a given time interval. The results are shown in Fig. 6. Obviously, it can be found that with the increase of the number of enactment service users, the average response time is increasing. However, in the case of SGWE, the average response time is much smaller than that in the case of single enactment service. And the gap between these two cases is increasing with the increase of the number of users, which shows the advantage of distributed workflow system.

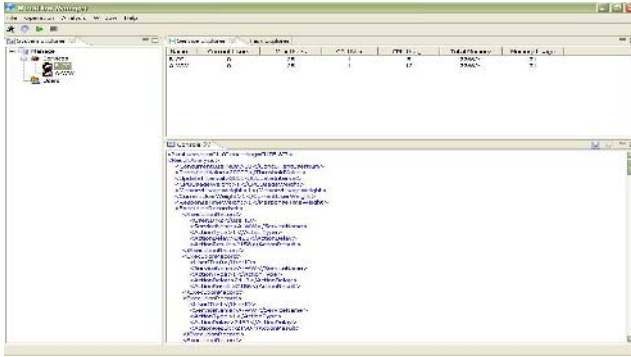


Fig. 5. A Management Tool of SGWE

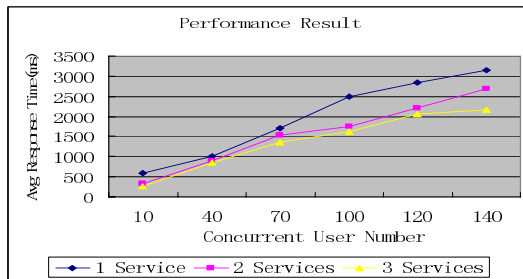


Fig. 6. Experiment Results of Distributed Grid Workflow Engine

5 Related Works

Since grid workflow plays a very important role in grid application, there are many ongoing grid workflow projects. In [5] some well-known grid workflow systems are classified according to their frameworks. But unfortunately, the structure of grid workflow engine is not included in this framework. As far as we know, most grid workflow systems employ a centralized structure, for example, GSFL [6], GridAnt [7] and Condor [8]. In [9], the structure of GeneGrid Workflow is introduced, when a new workflow instance is running, the workflow management factory will generate a workflow manager service instance for this workflow instance. Although workflow manager service instances can be distributed into different machines, this topic is not discussed in this paper.

There are already many distributed business workflow management systems in the market. Although the components of these systems are distributed, their structures can not be changed dynamically. In [10], a goal-driven autoconfiguration tool is introduced. This tool aims to recommend an appropriate system configuration in terms of replicated workflow, application, and communication servers, so as to meet given goals for performance, availability, and performability at low system costs. The basic idea of our system is very similar to theirs. In their system, workflow engine can have several copies and load is distributed to these engines. In our system, the interfaces of

workflow engine are implemented as separate services, which can be deployed independently. It increases the scalability and also flexibility of the whole engine. Besides this difference, they constructed their tool based on CORBA, while in our system, Jini is adopted so that the system can be re-configured more easily with the help of Jini services. In [11], the load-balancing technology for distributed WFMSs is discussed and a workflow load index to measure load level of workflow engines is defined. But it did not discuss how to develop a re-configurable workflow engine.

6 Conclusions and Future Work

In grid environment, it is possible an application has tremendous concurrent users. Since workflow is a basic service in such environment, the heavy load will lead centralized workflow engine become a bottleneck. In this paper, we proposed a distributed re-configurable grid workflow engine. In this architecture, enactment services can be added or removed on demand. Therefore, the whole engine will have high performance and high availability.

Although we have implemented a prototype system, there are many functions waiting to be developed. Future work will include but not limit to:

- (1) To design an algorithm to calculate the load brought by each request so that we can allocate this request to a service with corresponding capability;
- (2) To design more efficient load-balancing algorithm, which can optimize the parameters itself;
- (3) To develop a self-deploying architecture. In this architecture, the engine can find the available resources and when the load is increased, it will deploy an enactment service to this resource. And when the load is decreased it can shut down this service automatically. We have developed the auto-deploying tool for services and we will integrate this tool into the system.

Acknowledgements

This research is supported by Chinese NSF Project (No. 60503041). This work is also partly supported by "SEC E-Institute: Shanghai High Institutions Grid", Chinese Semantic Grid Project (2003CB317005) and Chinese NSF Project (No.60473092).

References

1. Foster, I., Kesselman, C., Tuecke, S., The Anatomy of the Grid: Enabling Scalable Virtual Organization, *International Journal of Supercomputer Applications*, Vol. 15(3), 2001, pp200-222
2. Foster, I., Kesselman, C., et.al., The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *International Journal of Supercomputer Applications*, Vol. 15(3), 200-222, 2001

3. Yolanda Gil, Ewa Deelman, Jim Blythe, Carl Kesselman, Hongsuda Tangmunarunkit, Artificial Intelligence and Grids: Workflow Planning and Beyond, IEEE Intelligent Systems, 2004.1, pp26-33
4. Jini(TM) Architecture Specification. http://www.sun.com/software/jini/specs/jini2_0.pdf. Version 2.0 June 2003.
5. Jia Yu and Rajkumar Buyya, A Taxonomy of Workflow Management Systems for Grid Computing, Technical Report, GRIDS-TR-2005-1, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, March 10, 2005
6. Krishnan, S., Wagstrom, P., von Laszewski, G.: GSFL: A Workflow Framework for Grid Services. Technical Report, The Globus Project, <http://www-unix.globus.org/cog/projects/workflow/gsfl-paper.pdf>, 2002
7. von Laszewski, G. K. Amin, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi, GridAnt: A Client-Controllable Grid Workflow System. In 37th Annual Hawaii International Conference on System Sciences (HICSS'04) IEEE Computer Society Press, Los Alamitos, CA, USA, January 5-8, 2004
8. Condor: The Directed Acyclic Graph Manager. <http://www.cs.wisc.edu/condor/dagman/>, 2003
9. David R. Simpson, PV Jithesh, Noel Kelly et al, GeneGrid: A Practical Workflow Implementation for a Grid Based Virtual Bioinformatics Laboratory, Proc. of the UK e-Science All Hands Meeting 2004 (AHM04), 2004, pp. 547-. 554
10. Michael Gillmann, Jeanine Weissenfels, German Shegalov, Wolfgang Wonner, Gerhard Weikum, A Goal-driven Auto-Configuration Tool for the Distributed Workflow Management System Mentor-lite, <http://www.mpi-sb.mpg.de/departments/d5/software/mlite/papers/Demo-SIGMOD-submitted.pdf>, 2004
11. Li-jie Jin, Fabio Casati, Mehmet Sayal, Ming-Chien Shan, Load balancing in distributed workflow management system, Proceedings of the 2001 ACM symposium on Applied computing 2001, Las Vegas, Nevada, United States pp 522 - 530

Adding Instruments and Workflow Support to Existing Grid Architectures

D.J. Colling¹, L.W. Dickens¹, T. Ferrari², Y. Hassoun¹, C.A. Kotsokalis³,
M. Krznaric¹, J. Martyniak¹, A.S. McGough¹, and E. Ronchieri²

¹ Imperial College London

{d.colling, luke.dickens, y.hassoun, marko.krznaric, janusz.martyniak,
andrew.mcgough}@imperial.ac.uk

² INFN, Italy

{Tiziana.Ferrari, Elisabetta.Ronchieri}@cnaf.infn.it

³ GRNET, Greece

ckotso@grnet.gr

Abstract. Many Grid architectures have been developed in recent years. These range from the large community Grids such as LHG and EGEE to single site deployments such as Condor. However, these Grid architectures have tended to focus on the single or batch submission of executable jobs. Application scientists are now seeking to manage and use physical instrumentation on the Grid, integrating these with the computational tasks they already perform. This will require the functionality of current Grid systems to be extended to allow the submission of entire workflows. Thus allowing the scientists to perform increasingly larger parts of their experiments within the Grid environment. We propose here a set of high level services which may be used on-top of these existing Grid architectures such that the benefits of these architectures may be exploited along with the new functionality of workflows.

1 Introduction

As the Grid is becoming more significant to the application scientist, they now seek to perform ever more complex work with it. Initially they have sought to run computationally expensive executables over the Grid. Hence the Grid has evolved (in general) as a batch submission system (for example [1, 6, 14, 18]). However, as their confidence has grown with the Grid, they now seek to orchestrate the whole of their scientific process on the Grid. This will often require use of scientific equipment along with the storing and/or processing of instrument output. This process may be repeated many times before the scientist obtains the results that they desire. The way in which these tasks are performed and the way in which they interact with each other can be described in terms of a workflow.

By adding instruments into the Grid we not only increase the requirements on the Grid to support workflows but also increase the requirement for many other functionalities. Many of these are well known to the Grid community, although in general these are not widely supported yet.

Workflows are not a new concept to the application scientist. Many have their own ad-hoc ways of performing workflows. In general these often require much human intervention, thus they are eager to automate this process as much as possible. However, the automated processing of workflows through a Grid architecture needs to proceed in a timely manner and match any Quality of Service (QoS) requirements the application scientist may have. To achieve this, reservations become important, both for the instruments and the existing Grid resources. Instruments are often in high demand, thus the ability to reserve time on them becomes significant. Conversely when an experiment is run on such an instrument, it is vitally important that the Grid resources are available to cope with the results produced. In order to make reservations it is also essential to provide a framework for the establishment of *Service Level Agreements* (SLAs).

In order to provide workflow support within the Grid it is possible to develop a completely new architecture from the ground up. Although this does have the advantage that the architecture can be defined to be workflow aware from the outset, it does have the disadvantage of not reaping the benefits already achieved from the existing Grid middlewares, such as wide acceptance within the community or robustness evolved from community exposure and hardening.

In this paper we propose how to augment existing Grid middlewares with workflow support. Through this we will use as much as possible of the existing middleware, such as the ability to load-balance and schedule tasks or monitor the running of these tasks. Workflow functionality is thus added as a higher level service on-top of the existing infrastructure.

We present in section 2 an overview of related work in this area. Section 3 describes the higher level services required to provide workflow and instrument support to the existing Grid architectures while section 4 provides more specific details for our work within the GRIDCC project. We conclude in section 5 and describe how we intend to progress with this work.

2 Related Work

Many Grid workflow systems and tools exist, such as: Askalon [9], DAGMan [16], GridFlow [4], Gridbus Workflow [23], ICENI [11, 13], Pegasus [5], and Triana [17]. Yu and Buyya [24] present a detailed survey of existing systems and provide a taxonomy which can be used for classifying and characterising workflow systems. We use elements of the taxonomy related to Quality of Service (QoS) and workflow modelling for comparing our approach to other systems.

A prominent feature of adding instruments to the Grid is the need for real-time remote control and monitoring of instrumentation. Users and applications will be allowed to make *Advance Reservation* (AR) of computational resources and instruments. AR guarantees availability of resources and instruments at times specified [15]. In ICENI, the scheduling framework supports AR by using the meta-data of application components together with corresponding historical data stored for the purpose of performance analysis and enhancement [12, 21, 22]. This approach is also used in Pegasus, Gridbus Workflow and Askalon. Some

systems like Askalon, DAGMan, ICENI, GridFlow and Gridbus Workflow allow users to define their own QoS parameters and to specify optimisation metrics such as application execution time, desired resources and economical cost. Other systems such as Pegasus and Triana do not. We are using a similar mechanism for performance enhancement and estimation taking into account the real-time QoS constraints.

3 The Architecture

In this section we outline a generic architecture which may be applied to many of the existing Grid architectures. Figure 1 illustrates this generic architecture for providing workflow and instrument support. The rest of this section outlines the requirements for each of the items in this generic architecture and the functionality they should provide.

Planner: The planner sits between the user environment and the Grid providing a stable interface for both application scientists and computer experts.

The planner accepts complex abstract workflows from the user describing an application which needs to be executed along with any QoS requirements. QoS may include requirements on total (or partial) runtime and/or maximum price. When writing workflows, the user should generally not be concerned with choosing specific resources, but rather with resource/service types. This not only hides unnecessary detail, but also allows writing portable workflows for submission at different sites and different times.

User-specified high-level constraints are then decomposed by the planner in terms of a simple set of basic pattern behavior [20]. In order to draw a quantitative picture of the system, the planner extracts performance parameters about resources and then independently analyzes the information to produce a concrete set of requirements. Equipped with the above parameters from the available resources, the planner builds an optimal workflow and deploys it using a workflow engine.

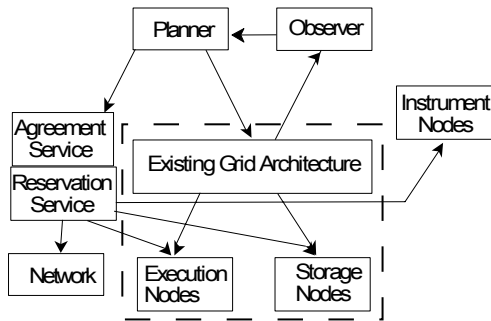


Fig. 1. The generic Architecture

Observer: In an ideal world once the planner has finished its task the workflow can be deployed successfully to the selected resources and run as desired. However, the nature of the Grid is that of an environment where resources may appear and disappear without warning, network links break and tasks not perform as expected. This may be both to the advantage of a application scientists workflow as well as a disadvantage.

The observer is designed to monitor not only the progress of the executing workflow but also the state of the Grid. If the workflow is not operating as expected (both in a good or a bad sense) then it can prompt the planner to alter the workflow plan. Alternatively if the observer discovers new more appropriate resource it can prompt the planner to make use of these.

Reservation Service: Advance reservation is one of the mechanisms used to support reliable service levels, for a specific time interval in the future, by a set of resources satisfying specific requirements. This mechanism is useful to various mission-critical applications, for example to Grid applications for the reservation of network bandwidth for bulk transfers, to Data Scheduling services requiring storage space reservation prior to the start of a data transfer session, and to workload management services to allocate computing power (e.g., to high-priority execution tasks).

Instrument Node: The Instrument denotes a coherent collection of services that provide all the functionality for controlling and monitoring of a physical instrument. The main features of IE are:

1. Accessibility through standard interfaces.
2. Allowing for remote controlling and monitoring of physical instruments.
3. Providing exception handling mechanisms with very low latency which, in case of locally thrown exceptions, are capable of protecting physical instruments from potential harm.
4. Supporting information and monitoring services to insure effective and successful operation of the IE.

Existing Grid Architecture: has the role of allocating resources in such a way that concurrent user requirements are efficiently satisfied. Assigning resources with the aim to achieve optimal performance is known as the problem of resource allocation or *scheduling*. Globus [14], Condor [10, 16, 19], gLite [7] and Sun Grid Engine [8] are examples of middleware offering these capabilities. Typically they will abstract the underlying grid into **Execution Nodes** and **Storage Nodes**.

Security: As a remote, distributed and multiuser system, an interactive GRID is subject to a wide variety of threats. In order to eliminate unauthenticated and unauthorized access to the system a mechanism of secure information exchange has do be developed. In general this is best achieved through the use of certificates, based on user credentials, thus allowing for single sign-on to the Grid. Users can then be first authenticated before checking to see if they are authorised to use the particular service. In general non-symmetric key cryptography is preferred due to its higher level of security. Though as interaction with instruments may be time critical symmetric keys may be needed.

There are potential solutions available to achieve above requirements. The widely used X.509 certificates might be utilised to allow secure resource access, and when needed, a faster symmetric Kerberos cryptography system. A single sign-on might be realised by using proxy certificates.

Agreement Service: In order to make use of reservations within the Grid along with the ability to draw up SLA's it is required to have some form of service which can perform this task. This is carried out by the agreement service. This service is capable of making multiple agreements between different Grid resources which may or may not be concurrent.

4 Example Implementation Case: GRIDCC

Grid Enabled Remote Instrumentation with Distributed Control and Computation (GRIDCC) is a European project that aims at extending the current Grid technologies which provide batch access to distributed computational and storage resources, so these technologies include access to and control of distributed instrumentation.

4.1 Use Cases

A number of Grid workflow use cases involving the interaction with instruments have been identified in the framework of the GRIDCC project [2]. In this paper we focus on on distributed intrusion detection and meteorological conditions prediction. Other applications based on workflows involving the interactions with instruments are the high energy physics and power Grids.

Intrusion detection is applied to network infrastructures with the aim to detect *Denial of Service* (DoS) attacks. In this case Instrument Elements (IEs) are services responsible for collecting and processing large amounts of data gathered by network routers. The corresponding workflow is relatively simple as it contains one decision point and one endless loop. Firstly, data collected by routers is processed by the IE, which performs data collection, sampling, aggregation and filtering. Then, results are handed to a soft computing component which performs additional processing to reduce the volume of collected data. Finally, data is fed to a central decision-support system and a decision is made. If no attack is detected, control is returned to the IE in order to let the processing continue, otherwise the following tasks are performed in the system: processing takes place to back-trace it; network devices are reconfigured accordingly; the "reaction" phase of the workflow ends, while traffic processing on the IE continues.

The meteorological use case concerns an application with the aim to use meteorological data for whether prediction and alerting when extreme whether conditions are expected. The meteorological workflow is more complex, as it includes a variety of Grid tasks and workflow patterns, such as advance reservation, multiple decision points, branched output, nested loops, etc.

The upper-bounded time requirements of these two applications have to do with the requirement to react urgently under an unusual situation: The presence of an attack, or probabilistic model results indicating extreme whether conditions.

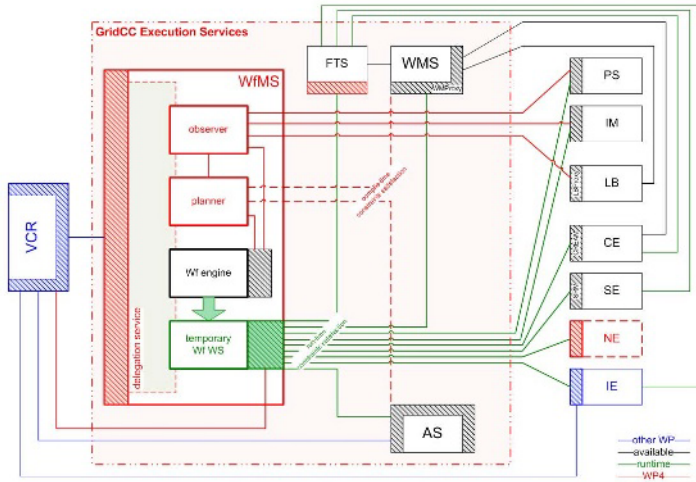


Fig. 2. The GRIDCC Architecture

4.2 The GRIDCC Architecture

In this section, we give an account of the GRIDCC architecture. To give the GRIDCC components relevance, they must be placed in context with an existing grid toolbox; here the gLite [7] model is used. Figure 2 shows the overall structure of this architecture. More details are available in [2]. The GRIDCC architecture represents a realisation of the abstract architecture depicted in Figure 1. The GRIDCC architecture constitute a set of components each of which represents a set of services. There are services for planning and submitting workflows, services for maintaining and monitoring the tasks that make up the workflow, agreement services for reserving resources, for monitoring and controlling physical instruments and services dealing with security issues. GRIDCC uses the EGEE-gLite packages [7] where the main components of the GRIDCC architecture are:

The Virtual Control Room (VCR) which represents the user interface that enables users to build and submit complex workflows.

The Instrument Element (IE) which represents a novel concept to the GRIDCC project. A Local Problem Solver (LPS) is embedded within IE for diagnosing and reacting to error conditions in such a way as to protect the physical instrument from harm. Also, a performance Information and Monitoring Service (IMS) is provided to ensure LPS reliability and successful operation of the IE.

Compute and Storage Elements (CE & SE) are similar to those which exit within other Grid projects.

The Execution Services (ES) which are responsible for the execution of the user-defined workflows and maintaining the status of the tasks that make up the workflow. The ES include: the Workflow Management System (WfMS) for workflow management – which is decomposed here into the Planner, the Observer,

a workflow engine (provided by a third party), and a temporary workflow Web Service – the Workload Management System (WMS) for logging and bookkeeping and service discovery, and the Agreement Service (AS) for reservation management. The File Transfer Service (FTS) is used for staging files to and from the Compute Elements as required.

The Agreement Service (AS) which is responsible of handling resource reservation requests and is invoked by the WMS. The AS controls the quality of service by making resource reservation agreements with the available resources, IEs, CEs or SEs. The AS attempts to satisfy a single resource reservation request by invoking one or more resource service providers.

The Problem Solver (PS) which may be manifested as either a Local Problem Solver (LPS) or a Global Problem Solver (GPS) is a new element included into the Grid in an attempt to protect the instruments. A Local Problem Solver will monitor a single (or small collection of) instruments in an attempt to prevent the instrument being damaged due to incorrect use from the Grid. The GPS performs the same task but deals with problems caused by collections of instruments.

Information and Monitoring (IM) service gathers information about all services currently active within the Grid.

Network Element (NE) is a service to represent the network within the Grid. This allows networking resources to be reserved.

5 Conclusions and Future Work

We have implemented the GRIDCC workflow architecture which is capable of taking a workflow deploying this to a workflow engine. The engine is then able to submit the stages of the workflow to the existing (in this case gLite) Grid middleware. We are now investigating techniques to manipulate the workflows that are submitted in order to better meet the QoS requirements of the system. We are looking into the techniques proposed in [3]. We are also developing the agreement and reservation capabilities.

References

1. Project SGE. <http://www.sun.com/software/gridware/>.
2. The GridCC Architecture Version 1.1.4. <http://www.gridCC.org>, 2005.
3. A. S. McGough and J. Cohen and J. Darlington and E. Katsiri and W. Lee and S. Panagiotidi and Y. Patel. An End-to-end Workflow Pipeline for Large-scale Grid Computing. Accepted for inclusion in the Journal of Grid Computing.
4. J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd. GridFlow: Workflow Management for Grid Computing. In *Proceedings of 3rd International Symposium on Cluster Computing and the Grid (CCGrid), Tokyo, Japan*. IEEE CS Press, Los Alamitos, 12–15 May 2003.
5. E. Deelman, J. Blythe, Y. Gil C. Kesselman, G. Mehta, K. Vahi, A. Lazzarini, A. Arbre, R. Cavanaugh, and S. Koranda. Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*, 1(1):9–23, 2003.

6. EGEE. Enabling Grids for E-science. <http://public.eu-egee.org/>.
7. eGee gLite Lightweight Middleware for Grid Computing. <http://glite.web.cern.ch/glite/>, 2005.
8. Sun Grid Engine. <http://www.sun.com/software/grid/>, 2005.
9. T. Fahringer, A. Jugravu, S. Pillana, R. Prodan, C. Seragiotto Jr, and H. L. Truong. ASKALON: a tool set for cluster and Grid computing. *Concurrency and Computation: Practice and Experience*, 17(2-4):143–169, 2005.
10. M. Litzkow, M. Livny, and M. Mutka. Condor - A Hunter of Idle Workstations. In *Proceedings of 8th International Conference of Distributed Computing Systems (ICDCS), Los Alamitos, CA, USA*, pages 104–111. IEEE CS Press, 1988.
11. A. Mayer, S. McGough, N. Furmento, W. Lee, S. Newhouse, and J. Darlington. ICENI Dataflow and Workflow: Composition and Scheduling in Space and Time. In *UK e-Science All Hands Meeting, Nottingham, UK*, pages 894–900. IOP Publishing Ltd, Bristol, UK, Sep. 2003.
12. S. McGough, L. Young, A. Afzal, S. Newhouse, and J. Darlington. Performance Architecture within ICENI. In *UK e-Science All Hands Meeting, Nottingham, UK*, pages 906–911. IOP Publishing Ltd, Bristol, UK, Sep. 2004.
13. S. McGough, L. Young, A. Afzal, S. Newhouse, and J. Darlington. Workflow Enactment in ICENI. In *UK e-Science All Hands Meeting, Nottingham, UK*, pages 894–900. IOP Publishing Ltd, Bristol, UK, Sep. 2004.
14. Globus Project. <http://www.globus.org>, 2005.
15. S. Andreozzi and T. Ferrari and S. Monforte and E. Ronchieri. Agreement-based Workload and Resource Management. In *Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing*, Melbourne, Australia, December 2005. IEEE Computer Society.
16. T. Tannenbaum, D. Wright, K. Miller, and M. Livny. *Condor - A Distributed Job Scheduler. Beowulf Cluster Computing with Linux*. The MIT Press, MA, USA, 2002.
17. I. Taylor, M. Shields, and I. Wang. Resource Management for the Triana Peer-to-Peer Services. In J. Nabrzyski, J. M. Schopf, and J. Weglarz, editors, *Grid Resource Management - State of the Art and Future Trends*, pages 451–462. Kluwer Academic Publishers, 2004.
18. Condor Team. Condor Project Homepage. <http://www.cs.wisc.edu/condor>.
19. D. Thain, T. Tannenbaum, and M. Livny. *Condor and the Grid. Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, NJ, USA, 2003.
20. W.M.P. van der Aalst. Workflow Patterns. <http://www.workflowpatterns.com/>.
21. L. Young and J. Darlington. Scheduling Componentized Applications on a Computational Grid. MPhil/PhD Transfer Report, Imperial College London, University of London, 2004.
22. L. Young, S. McGough, S. Newhouse, and J. Darlington. Scheduling Architecture and Algorithms within the ICENI Grid Middleware. In *UK e-Science All Hands Meeting, Nottingham, UK*, pages 5–12. IOP Publishing Ltd, Bristol, UK, Sep. 2003.
23. J. Yu and R. Buyya. A Novel Architecture for Realizing Grid Workflow using Tuple Spaces. In *Proceedings of 5th IEEE/ACM International Workshop on Grid Computing (Grid 2004), Pittsburgh, USA*. IEEE CS Press, Los Alamitos, 8 Nov. 2004.
24. J. Yu and R. Buyya. A taxonomy of workflow management systems for grid computing. GRIDS-TR-2005-1, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, March 10, 2005.

Workflow Deployment in ICENI II

A. Stephen McGough, William Lee, and John Darlington

London e-Science Centre
Department of Computing
Imperial College London, London
{asm, ww1, jd}@doc.ic.ac.uk

Abstract. The Imperial College e-Science Networked Infrastructure (ICENI) has been developed by the London e-Science Centre for over four years. ICENI has prototyped many novel ideas for providing an end to end Grid middleware. This has included: service-oriented architecture, component programming model, retaining and using meta-data collected throughout the life-cycle of an application, and scheduling algorithms which are aware of workflow and performance data. In this paper we describe the workflow pipeline and deployment process of ICENI II. This allows the user to specify their workflow at an abstract level which is fed through the pipeline in order to successfully deploy it over the Grid. The tool sets, which make up the stages of the ICENI II pipeline, are designed to be composable in an à-la-carte fashion. Thus allowing Grid developers to select only those components which are relevant for their work. When these tool sets are composed together they form the higher level services required to make the Grid useful to the end user.

1 Introduction

The ICENI (Imperial College e-Science Networked Infrastructure) [8, 3] has been developed as a service-oriented architecture for (workflow) component composition. Our focus in ICENI has three major elements: prototyping the services and their interfaces necessary to build a service-oriented middleware; developing an augmented component programming model for authoring reusable Grid applications; to explore the meta-data needed to enable effective machine-assisted and automated decision-making; and the development of higher level services required to make Grid infrastructure usable. ICENI II is a re-working of the original ICENI concepts using the newly evolving commodity tools, standards and services. Allowing us to re-focus on the higher level services required to make the Grid useful to the scientist. In this paper we present the workflow pipeline and deployment service for ICENI II focusing on the core services that can be composed to enable a (workflow) component programming paradigm.

In ICENI an *application* is a composition of components and services defined using a *workflow language*. A *component* is an indivisible unit of execution in which all the contextual, functional and behavioral aspects are made explicit. While multiple *components* might exhibit the same functional interface, they can be independently implemented yielding different behavior [10]. Furthermore, a

component is an abstract entity that awaits composition by *application authors* and subsequent deployment into an *execution environment*.

A deployed *component* is referred to as a *service*. A *service* is a realised manifestation of the states and functions provided by the underlying *resource* and the *component* implementation. The definition does not mandate a particular *component* implementation architecture. The current ICENI II implementation requires the use of an interoperable set of protocols (e.g. SOAP) and data model (e.g. WSDL, XSD Schema) as the common denominator. We envisage the protocol details are sufficiently abstract from the component and application authors. This allows *applications* to be composed with both abstract *components* that need to be provisioned and anchored *services* that exist irrespective of the lifetime of the *application*. It is the role of the *workflow pipeline* to realise an *application* by performing component implementation selection, resource allocation, deployment and orchestration. Further information on the ICENI component model can be found in [8, 9].

2 Architecture

Figure 1 illustrates the workflow pipeline for ICENI II. Within this design we have determined three main stages within an application’s life-cycle:

Specification of the application to be executed. At this stage the workflow may be abstract in nature - the code that will be used to implement the tasks and the

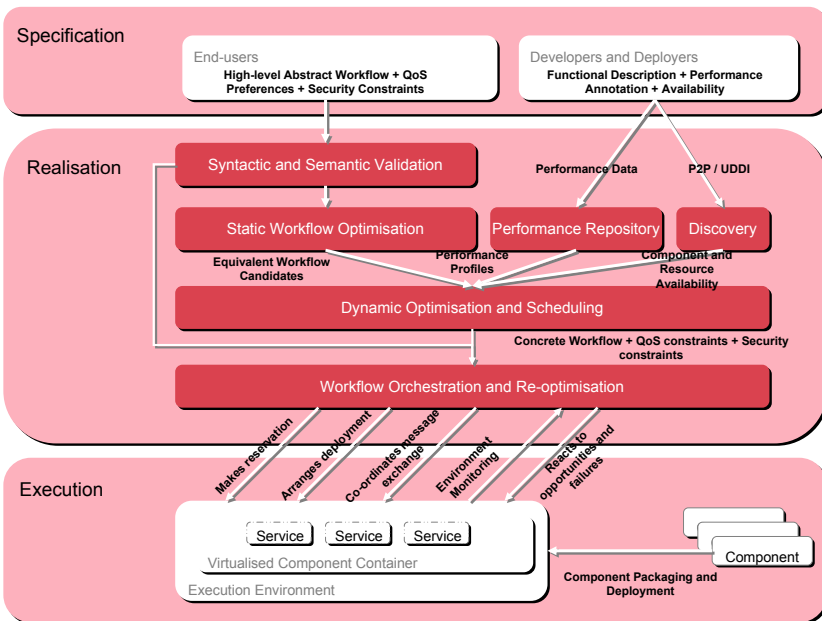


Fig. 1. The Design Diagram for ICENI II

resources on which they run are not yet defined. The translation of an application scientist's ideas into these workflows is a complex task. It cannot be assumed that the application scientist will be an expert with workflow languages, we therefore see the need to develop application specific interfaces which allow the scientist to describe their problem in their own space - translating this down into a workflow. This is an area of active research. We are working with the bio-informatics community within the e-Protein project [13]. The Taverna project [21] is a good example of this process.

Realisation of the workflow. The workflow needs to be made concrete before it can be executed. This process involves determining the appropriate resources and software (component implementations) dependencies for the workflow to operate. The Realisation stage of the life-cycle is discussed further in Section 3. **Execution** of the workflow. In this stage the concrete workflow is enacted on the defined set of resources. Implementations will be made active on resources at the appropriate point in time. If however the workflow does not proceed as planned then the realisation stage can be re-entered so that the workflow may be altered in light of the changes. It is also possible to re-enter the realisation stage in cases where parts of the original workflow couldn't be made concrete at the original pass or when new opportunities become available; such as availability of new resources. Further details of the Execution stage are given in Section 4.

Once a workflow enters the realisation stage an *Application Service* will be designated to represent the running application. It acts as a central point where the user can find out about the state or influence the progress of the application.

3 Realisation of Workflows

The Realisation stage selects the best use of anchored services or resources and component implementations available on the Grid in an attempt to match the criteria specified by the application scientist. Most often this will be in terms of time constraints that the scientist may have. Although, other constraints may be used (or combined) such as cost of resources, reliability of execution or the level of trust required from the used resources and components. This is often referred to as a Quality of Service (QoS) Document. This stage will take the abstract workflow and resolve it into a concrete workflow in which implementations and resources will have been determined.

The Realisation stage can be decomposed into several complementary services with most being optional. The only non-optional element, referred to as the *Resolver*, matches abstract components with anchored services or implementations and resources. Pluggable optimisers can be used to improve the performance of the workflows or matching of QoS requirements. Detailed in the following subsections are the pluggable elements that we have defined so far [11].

Syntactic and Semantic validation. On entering the Realisation stage the workflow is checked for syntactic and semantic validity. The first step in checking syntactic correctness ensures that all required inter-component connections have

been made, i.e. all ports that must be connected are, otherwise the workflow will be unsuccessful at a later stage. The second step is to verify that the data to be exchanged through each of these connections is a valid transaction.

The workflow may now be checked for semantic validation. Scientific knowledge about what action a component performs, and what the meaning of its inputs and outputs, can be used to determine if the connection of two components makes sense. For example it may be syntactically correct to feed a matrix into a finite difference solver, though if this matrix is a diagonal solution to a set of linear equations this makes little semantic sense. If a workflow fails at a syntactic or semantic level then it is returned to the specification stage.

Static Workflow Optimisation. The Workflow Static Optimisation Service is responsible for manipulation of the workflow. Using static information about the components and their composition this service generates a new workflow which is expected to execute more efficiently than the original. This is achieved through such steps as re-ordering of the components to improve efficiency; inserting additional components to allow particular implementations to communicate with each other; substitution of semantically equivalent workflow sections; pruning of redundant components from composed workflows; and the substitution of alternative implementations of components. It should be noted that the Workflow Static Optimisation Service does not consider the dynamic load on system's within the Grid.

Pruning of Resource Space. Once the workflow has been optimised it is now essential to reduce the set of resources that will be considered. This is achieved by looking at the slowly changing resource information such as authorization to use a resource; hardware / software requirements; problem specific requirements such as long execution time with no option to checkpoint; and sensibility of selection.

Workflow-aware, performance-guided scheduling. The aim of most schedulers is to map the abstract workflow to a combination of resources and implementations that is both efficient in terms of execution time of the workflow and in terms of the time to generate the concrete workflow. Components need not all be deployed at the same time: just in time scheduling of components and the use of advanced reservations help to make more optimal use of the available resources for both the current and other users.

Schedulers need to be designed to be workflow aware. A number of workflow aware schedulers have been prototyped in ICENI [12] and we are now developing new workflow aware performance guided schedulers using constraint equations solved using Mixed Integer Linear Programming [15]. Thus the scheduling of components depends not only on the performance of a component on a given resource, but also on the effect this will have on the other components in the workflow. Described below are the general steps taken to evaluate a suitable mapping of components onto resources.

The scheduler can speculatively match implementations with resources. The scheduler can then interrogate performance information in order to obtain estimates on the execution times for these implementation / resource combinations

along with any implications for the overall workflow. With this information and information gathered from the resources that have been discovered, the scheduler can determine an appropriate mapping of the components over the resources.

4 Execution of a Workflow

Due to the uncertainties of resource and network availability in a dynamic system such as the Grid, it is necessary to support advanced reservations to provide QoS guarantees. Reservations may be made on computational resources, storage resources, instruments or the underlying fabric of the Internet such as network links. The reservations may be made for exclusive use of the entity or, in some cases, some pre-agreed portion of it. This can be realised through the introduction of market forces into the Grid [6].

The execution environment represents the virtualisation of the resource that manages the life-cycle of the parts of an application (see Figure 2). The execution environment encapsulates the facilities available to the software component, such as inter-component communication, logging, monitoring, failure recovery, checkpointing and migration. These facilities are exposed to the software component through a set of abstract APIs. These abstractions allow the execution environments managing the parts of an application to co-operate and co-ordinate their runtime capabilities, such as network transport, co-location and shared file system. Software engineers developing the components are insulated from the implementation choice made by the optimisation stage by following the software patterns offered by the APIs. This is analogous to the MPI[5] abstraction for message-passing in parallel applications.

The software component instantiated in the execution environment is referred to as a service. We adopt Web Services as one *view* of the running software component. It is an ideal way for services on different physical resources to communicate with each other in an interoperable manner. The elements in the execution environment will be discussed in more detail.

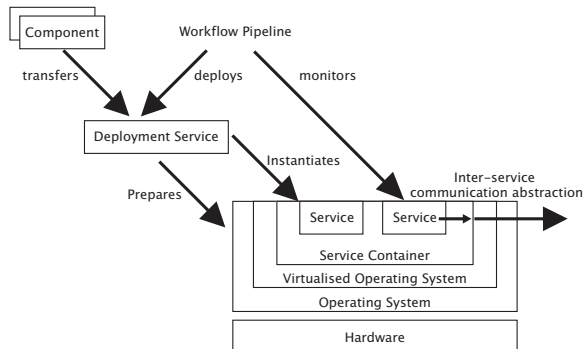


Fig. 2. Execution Environment and Multi-level Virtualisation

4.1 Component Deployment

A deployment service is the gateway to a computational resource. It is responsible for facilitating the provisioning and instantiation of a component assigned to a particular resource. Firstly, the deployment service prepares the execution environment. This might involve the preparation of a component container in a cluster resource. Recent advances in virtualisation technologies [1, 20] offer operating system-level virtualisation. Within the virtualised operating system, a component container provides the higher-level abstraction to the software component on top of the operating system facilities. The compartment model offers attractive features such as security and fault isolation. Multi-level virtualisation allows runtime facilities to be flexibly configured depending on the deployment requests [16]. Although virtualisation provides a sandbox environment for a component to execute seemingly exclusively, the cost in instantiating the container on-demand [7] may be too high for short-running components. Predictive instantiation might alleviate the setup cost by allocating resources in advance.

Once an execution environment is available, the deployment service will facilitate the provision of the software component onto the resource. This might involve the staging of software packages and their dependencies available remotely into the system. In order for this architecture to succeed across the Grid, a standardised interface for deployment and a language for software requirement description is essential. It reduces the need for users and software agents to understand a large number of description languages and deployment mechanisms to exploit a variety of Grid resources.

GridSAM [4] is a job submission and monitoring Web Service developed through collaboration with the OMII [14]. GridSAM allows the deployment of legacy code through a standard interface and is seen as one of the first toolkit items within ICENI II.

4.2 Checkpointing and Migration

Checkpointing is a technique for preserving the state of a process in order to reconstruct it at a later date. It is a crucial element for providing fault-recovery from a saved state. In scientific applications checkpointing provides a means for long-running simulations to be restarted at a previously examined parameter space [2]. This is also an important means for migrating the state of a process to another execution environment. This is often triggered by a re-scheduling decision as in some distributed resource managers. Many checkpointing and migration systems exist including OpenMosix [18], OpenSSI [19], and Kerrighed [17].

Migration might occur as a result of a recovery operation after a host failure. Alternatively migration may be as the result of a desire to exploit new possibilities within the Grid, such as a new resource becoming available.

It is worth pointing out that not all components can be checkpointed. Techniques to checkpoint components are outlined below. The generality of the approaches increases from one to the next. Checkpointing and migration schemes can be classified into three broad categories. Application-level checkpointing

might be initiated by the application itself through a checkpointing and migration API. This provides fine-grain control to the developer to save the states of the application at a critical moment of the execution, however it requires existing applications to be modified to take advantage of the functionality. Existing executables could also be made checkpointable by linking to checkpoint libraries that capture running stack, heap and program counter information in order to reconstruct the process remotely. This solution produces a checkpoint image that is rarely portable and complete because network sockets or file handles are inherently difficult to reconstruct. System level checkpointing provided by many virtual machine technologies allows the whole virtualised environment to be checkpointed. This provides a generic solution for most cases but the coarse nature means the checkpoint image would be very large.

In cases where the services within the workflow are capable of being checkpointed and migrated, the application co-ordinator in the workflow pipeline may re-schedule the service state to a suitable resource in order to respect the stated quality constraints. This might involve re-scheduling other services to different resources to achieve an optimal schedule. The stored checkpoints of services are transferred to the suitable resources through the deployment service and restarted in a reinstated execution environment. A component might receive messages during the time elapsed between its failure and restart. Such events are taken care of by the messaging abstraction in the initiating execution environment. The execution environment reports any anomalies like resource overloading or network failure to the application co-ordinator which in turn might trigger the migration process.

5 Conclusion

In this paper we have outlined a set of high level tool sets for the manipulation of workflows. When composed together these tool sets form the stages of a workflow pipeline which take a users abstract workflow through to deployment on Grid resources.

We are now developing some of these higher level services on top of GridSAM for job deployment and monitoring. Along with developments in the other services for optimising workflows across multiple resources.

Wherever possible, and appropriate, ICENI II is being developed using existing standards. To this end ICENI II will be developed as a set of Web Services and using up and coming standards such as BPEL4WS and JSDL.

References

1. P. Barman, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP 2003*, September 2003.
2. J. Chin, J. Harting, S. Jha, P.V. Coveney, A. R. Porter, and S. M. Pickles. Steering in computational science: mesoscale modelling and simulation. *Contemporary Physics*, 44:417-434, 2003.

3. N. Furmento, A. Mayer, S. McGough, S. Newhouse, T. Field, and J. Darlington. ICENI: Optimisation of Component Applications within a Grid Environment. *Journal of Parallel Computing*, 28(12):1753–1772, 2002.
4. Grid Submission and Monitoring service (GridSAM). <http://www.lesc.imperial.ac.uk/gridsam>.
5. W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828, September 1996.
6. Jeremy Cohen and John Darlington and William Lee. Payment and Negotiation for the Next Generation Grid and Web. In *UK e-Science All Hands Meeting*, Nottingham, UK, sep 2005.
7. K. Keahey, K. Doering, and I. Foster. From Sandbox to Playground: Dynamic Virtual Environments in the Grid. In *5th IEEE/ACM International Workshop on Grid Computing*, November 2004.
8. A. Mayer, S. McGough, N. Furmento, J. Cohen, M. Gulamali, L. Young, A. Afzal, S. Newhouse, and J. Darlington. *Component Models and Systems for Grid Applications*, volume 1 of *CoreGRID series*, chapter ICENI: An Integrated Grid Middleware to Support e-Science, pages 109–124. Springer, June 2004.
9. A. Mayer, S. McGough, N. Furmento, W. Lee, S. Newhouse, and J. Darlington. ICENI Dataflow and Workflow: Composition and Scheduling in Space and Time. In *UK e-Science All Hands Meeting*, pages 627–634, Nottingham, UK, September 2003. ISBN 1-904425-11-9.
10. A. Mayer, S. McGough, M. Gulamali, L. Young, J. Stanton, S. Newhouse, and J. Darlington. Meaning and Behaviour in Grid Oriented Components. In *3rd International Workshop on Grid Computing, Grid 2002*, volume 2536 of *Lecture Notes in Computer Science*, Baltimore, USA, November 2002.
11. A. Stephen McGough, Jeremy Cohen, John Darlington, Eleftheria Katsiri, William Lee, Sofia Panagiotidi, and Yash Patel. An End-to-end Workflow Pipeline for Large-scale Grid Computing. *Journal of Grid Computing*, pages 1–23, February 2006.
12. S. McGough, L. Young, A. Afzal, S. Newhouse, and J. Darlington. Workflow Enactment in ICENI. In *UK e-Science All Hands Meeting*, pages 894–900, Nottingham, UK, sep 2004.
13. A. O’Brien, S.J. Newhouse, and J. Darlington. Mapping of scientific workflow within the e-protein project to distributed resources. In *Proceedings of the UK e-Science All Hands Meeting 2004*, Nottingham, September 2004.
14. Open Middleware Infrastructure Institute (OMII). <http://www.omii.ac.uk/>.
15. Yash Patel, A. Stephen McGough, and John Darlington.
16. E. Smith and P. Anderson. Dynamic Reconfiguration for Grid Fabrics. In *5th IEEE/ACM International Workshop on Grid Computing*, November 2004.
17. The Kerrighed project. <http://www.kerrighed.org/>.
18. The open Mosix project. <http://openmosix.sourceforge.net/>.
19. The open SSI project. <http://openssi.org/index.shtml>.
20. User Mode Linux. <http://user-mode-linux.sourceforge.net/>.
21. The Taverna Project Website. <http://taverna.sourceforge.net/>.

Agent-Based Middleware Architecture for Workflow in Grid Portals

Sangkeon Lee¹, Jaeyoung Choi¹, and Keumwon Cho²

¹ School of Computing, Soongsil University,
1-1 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea
seventy9@ss.ssu.ac.kr, choi@ssu.ac.kr

² Supercomputing Center, Korea Institute of Science and Technology Information,
52 Eoeun-dong, Yuseong-gu, Daejeon 305-306, Korea
ckw@kisti.re.kr

Abstract. In this paper, we propose an agent-based middleware architecture for the workflows used in Grid portal. We combined workflow model, workflow management system, job distribution, parameter scheduling, and service interface in the middleware architecture. The workflow model consists of three layers. The agent-based middleware architecture consists of five agents and three types of communication protocols. Users can design a lightweight, flexible and effective middleware, which can be used to construct a workflow-based Grid portal.

1 Introduction

Grid portals[1][2] are widely used to construct virtual scientific research environments to integrate various types of software such as simulation software, analysis software and visualization software. A scientific research may require iterative execution of the experiments with experimental data or change of parameters. The number of iteration can be larger than billion times. To process billions of experiments on computing resources, Grid computing is necessary. Scientists design their research with workflows[3][4], which integrates the software and experimental data with a flow of research scenario. Therefore, virtualization, resource scheduling[5], data management, security management, information integration, ontology processing[6], high-speed network, and other facilities should be integrated to construct a Grid portal.

However, construction of e-Science environments using Grid portal is not simple because Grid computing requires setting and managing tens of middleware and integrating them as a single system. The most difficult problem is that those middleware are designed separately, so integrating the heterogeneous middlewares causes many troubles.

Therefore, integrating complex layers between Globus Toolkit[7] and user interface into a single design is required and design of the integrated system must take care of all matters such as its workflow model, workflow management, job distribution, parameter scheduling[8], and the service oriented architecture.

In this paper, we propose an agent-based middleware architecture for workflow design in e-Science environments. This architecture uses a service-oriented workflow model, which is developed by our prior research, called Meta Services[9], which integrates functionalities mentioned above. Meta Services provide an approach to abstract and map a workflow for a service by overriding workflow’s attributes with the service’s parameters.

By combining Meta Services with an agent-based middleware architecture, we designed and implemented a lightweight, flexible, and effective middleware architecture which can be used to construct workflow-based Grid portals.

2 Agent-Based Middleware Architecture for the Workflow Model

The Agent-based middleware architecture is shown in Figure 1. It consists of five major agents : AM (Access Manager), OM (Ontology Manager), SM (Service Manager), RM (Resource Manager), and EM (Execution Manager). Our workflow model has three layers: SM on service layer, RM on flow layer, and EM on task layer. Description of services, flows, and tasks is stored as XML files and managed by OM. Finally, AM is required to handle security such as authentication and access control.

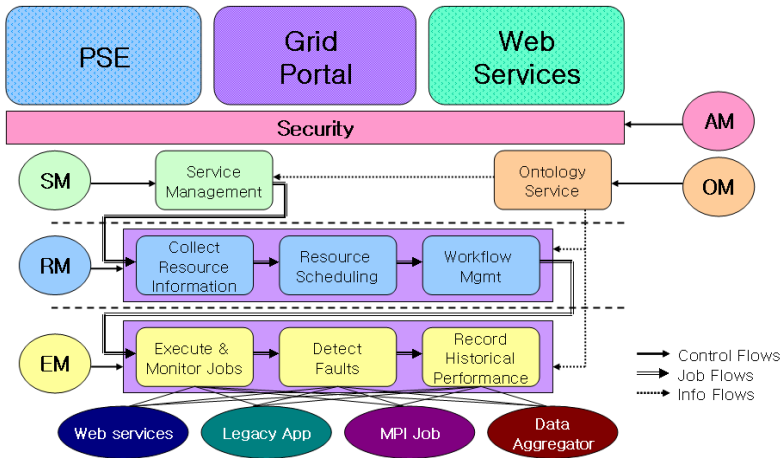


Fig. 1. Agent-based Middleware Architecture

2.1 Communication Among Agents

There are three types of communication protocols in the agent-based middleware architecture: admin protocol, user protocol, and agent protocol. An administrator has an administrator’s authority, and he can connect or manage SM, RM, and EM. Users can connect to OM through a GUI editor, and they can browse, search, and compose their works such as tasks, flows, and services depending on

their authorities. After a workflow scenario is prepared as a service by searching or composing, then a user submits the scenario to SM. Then, SM processes the overriding of workflow parameter and submits the changed flow to RM. After this process, RM analyzes the flow and distributes tasks to EMs, which connected to the RM. If a task is allocated to an EM, then the EM executes the task when a processor is available. Execution of flow and task is also performed with a user's authority, and it doesn't interrupt other user's process.

2.2 Distribution of Flow and Task

In our agent-based middleware architecture, an EM has a parameter named 'number of processing element'. Its value reflects how many tasks can be executed on the EM at the same time. By default the value sets with a number of processor in which the EM is installed. An administrator can change it using SET_PE operation. EM can also cooperate with other Grid job distributors by reducing its concurrency using SET_PE operation. EM has a waiting queue whose size is multiple of processing element. When an EM is started, it registers itself to a RM, and the RM maintains a list of EMs with information such as the number of processors and the size of waiting queue.

Each RM has a priority queue which queuing list of users connected to AM. When a user is appended to the list, a flow queue is created for the user. If an EM's waiting queue is freed, RM fills the EM's waiting queue with one of the activatable tasks from the flow queue of a user who has the highest priority. After the task is finished, the task owner's flow queue updates the list of activatable tasks.

2.3 Parameter Scheduling

Instead of appending prefix or postfix to data or parameter, parameter scheduling is performed by separation of tasks' working directory. When a task is executed, EM creates a temporary working directory `/msftmp/<username>/<flow>/<username>/<task id>` under a local user's home directory permitted by the grid-map file of hosts where the EM is executed. RM allocates a flow ID to the flow when it is submitted and task ID is assigned in a flow description stored in OM. When the task is completed its execution, the data file marked as an outputfile in the flow description is transferred to the next task's working directory. Therefore, even if a user doesn't override the name of an input or an output file, file corruption by conflict file names never occurred.

3 Conclusion and Future Work

In this paper, we present an agent-based middleware architecture for workflow-based Grid portals. The architecture contains essential functionalities for Grid portals. The functionalities include workflow, service interface, job distribution, and parameter scheduling. Our agent-based architecture consists of five agents and each agent has its own role over the corresponding workflow model. By using

this middleware model, users can design a lightweight, flexible, and effective middleware which is used to construct workflow-based Grid portals.

We have remodeled our pervious middleware MSF[10] using the agent-based middleware architecture and have implemented a prototype. We tested the middleware with simple scenarios which are used in Bio-grid portal for drug discovery. Workflow editor, management console, more complex workflow pattern, and performance optimization are required to use our middleware practically. So we are going to design and implement this features and continuously upgrade our middleware architecture.

Acknowledgement

“This work was supported by the National e-Science Project, funded by the Korean Ministry of Science and Technology (MOST).”

References

1. Allen, G., (ed.): Enabling applications on the Grid - a GridLab overview. *Intl. Journal on High Performance Computing Applications*. **17** 4 (2003) 449–466
2. Stevens, R., Robinson, A., Goble, C.: myGrid: personalized bioinformatics on the information grid. *Bioinformatics*. **19** 1 (2003) 302–304
3. Cao, J., Jarvis, S., Saini, S., Nudd, G.: GridFlow: Workflow Management for Grid Computing. 3rd International Symposium on Cluster Computing and the Grid. (2003) 12–15
4. Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M., Vahi, K.: Pegasus Mapping Scientific Workflows onto the Grid. In *Proceedings of across Grid EU Conference*. (2004)
5. Litzkow, M., Livny, M., Mutka, M.: Condor - A Hunter of Idle Workstations. 8th International Conference of Distributed Computing Systems. (1998) 13–17
6. Wroe, C., Stevens, R., Goble, C., Roberts, A., Greenwood, M.: A suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. *International Journal of Cooperative Information Systems special issue*. **12** 2 (2003) 197–224
7. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. *Intl. J. Supercomputer Applications*. **11** 2 (1997) 115–128
8. Abramson, D., Sosic, R., Giddy, J., Hall, B.: Nimrod: A tool for performing parametrised simulations using distributed workstations. In *Proc. of the 4th IEEE Symposium on High Performance Distributed Computing*. **8** (1995)
9. Lee, S., Choi, J.: Meta Services: Abstract a Workflow in Computational Grid Environments. *Lecture Notes in Computer Science*, Vol. 3516. Springer-Verlag, Berlin Heidelberg New York (2005) 916–919
10. Hwang, S., Choi, J., Park, H.: Meta Scheduling Framework for Workflow Service on the Grids. *Lecture Notes in Computer Science*, Vol. 3036. Springer-Verlag, Berlin Heidelberg New York (2004) 445–448

Cooperative Processes for Scientific Workflows

Khaled Gaaloul, François Charoy, and Claude Godart

LORIA - INRIA - CNRS - UMR 7503
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France
{kgaaloul,charoy,godart}@loria.fr

Abstract. The work described in this paper is a contribution to the problems of managing in data-intensive scientific applications. First, we discuss scientific workflows and motivate their use in scientific applications. Then, we introduce the concept of cooperative processes and describe their interactions and uses in a flexible cooperative workflow system called *Bonita*. Finally, we propose an approach to integrate and synthesize the data exchanged by the mapping of data-intensive science into *Bonita*, using a binary approach, and illustrate the endeavors done to enhance the performance computations within a dynamic environment.

1 Introduction

With the technological improvements and continuous increasing of scientific requirements, scientific applications are becoming complex, involving large-scale distributed environments and huge data exchanges. Since data processing can be costly, sharing these data products can save the expense of performing redundant computations [1]. The goal of scientific workflows is to provide an environment where scientists share their resources to pursue common goal. Indeed, the view that scientists typically perform experiments, where experiments can be considered as ordered collections of steps acting on data and involving a variety of distinct activities, motivates the exploitation of workflow technologies to scientific endeavors. In addition, the scientific domain specifies others requirements as the need to execute in a dynamic environment where resources are not known a priori [2]. To deal with those requirements, scientific workflows need more flexibility to cope with dynamic environments and more capabilities to manage large data streaming. However, the efforts done for the integration of distributed applications and heterogeneous data are still insufficient. Reusing analytical steps within a workflow instance involves an integration effort. Each analytic step consumes and produces data with a particular structural representation. To compose activities, the structural differences between the activities must be resolved, and this resolution is typically performed by the scientist either manually or by writing a special-purpose program or script [3]. Moreover, analytical steps used to be rigid and lack of flexibility. That is why, our research focuses on reducing those efforts by providing means, described in section 3, that facilitate the analysis and modeling of scientific applications.

The remainder of this paper is organized as follows. Section 2 briefly describes scientific workflows and motivates its use through an experimental example. Section 3 refers to the flexible cooperative workflow system *Bonita*, where we take a closer look to cooperative processes within *Bonita*. In section 4, we define our proposition for data-intensive science within our workflow system. In particular, we present a technique for mapping heterogeneous data within scientific workflows based on a binary approach. Section 6 concludes the paper and discusses some future works.

2 Preliminaries and Motivations

Traditionally, the scientific work is concentrated around experiments. Collecting, generating, and analysing large amounts of heterogeneous data is the essence of such work. The scientific experiments which have to be carried out during such a development process are typically composed of series of steps or tasks [4]. A global abstract view of a scientific work is that it consists of one or more steps with input and output data. There are many other aspects that scientific environments have to capture. These include mechanisms for cooperative work (e.g., ensuring the communication and the coordination for a group to work together in the pursuit of a shared goal) and data management (e.g., managing data format evolution in scientific experiments). Scientific projects can be considered as ordered collections of steps acting on data and involving a variety of distinct activities. So it motivates the exploitation of the workflow paradigm.

Scientific workflow is the application of workflow technology to scientific endeavours, and is recognized as a valuable approach for assisting scientists in accessing and analyzing data. Its application scopes are the support for large data flows, the need to decide on further steps after evaluating the previous ones [4], the need to monitor and control workflow execution including ad-hoc changes, and the need to execute in a dynamic environment where resources are not known a priori and may need to adapt to changes [2].

As an example, consider the development of prototypes by digitalization. We aim to develop, from an original model, a prototype that will be used for testing, measuring and controlling (e.g., metrology) the new model. First, we try to identify a number of features belonging to the original model by digitalizing it. Then according to the experience, which includes outcomes (measurements, cost aspects, etc.), we conduct another experiments (reconstruction and modification) to simulate it afterwards. If successful, the prototype will be ready to be tested depending on scientists requirements.

From a workflow point of view, this scientific experiment is typically composed of series of tasks, which are intertwined according to some control sequence (e.g., conditional branches), and receiving/producing manifold information. Thus, the scientific workflow seems to be an appropriate paradigm to integrate it.

3 Using BONITA for Scientific Workflows

Bonita is a flexible cooperative workflow system which incorporates the anticipation of activities as a flexible mechanism of a workflow execution. The principle of anticipation is to allow an activity to escape the traditional start-end synchronization model by ensuring intermediate results used as preliminary input into succeeding activities [5]. Our approach consists in adding flexibility to scientific workflow executions with minimal changes of the workflow model. We try to reach this goal by relaxing the way the model is interpreted. Scientists can take some initiatives regarding the way they start the assigned activities. The use of Bonita will ensure several advantages such as cooperative processes by ensuring numerous interactions between heterogeneous applications, the communication and the coordination for scientists to work together in the pursuit of a shared goal.

As described in section 2, we can ensure the flexibility of the scientific experiment. Actually the step of reconstruction and modification, called CAD task, is not completely finished. So we can anticipate it by passing to the next activities with the possibility of return. The anticipation in Bonita offers new alternatives in a dynamic environment. When the user selects an activity, he obtains the list of the executing activities: *executing* or *anticipating* states, and his assigned activities: *ready* or *anticipable* (see figure 1). Therefore, in our example, by anticipating the CAD’s step we will have the opportunity to optimize the process, to send a feedback to this step, and eventually add/remove an activity.

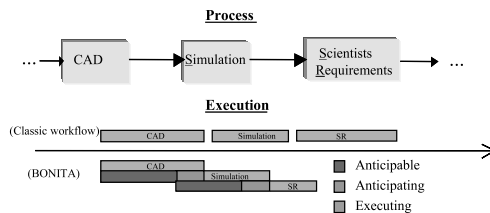


Fig. 1. Ensuring flexibility for a scientific workflow execution

4 Mapping Data-Intensive Science into BONITA

Bonita integrates a great number of services to control and simplify many cooperative aspects such as Java Message Service (JMS) to notify the definition and execution changes within a workflow process, and Jabber service that allows the users to receive notifications at real-time and exchange different kinds of messages. However, those services lack of heterogeneity and dynamic changes, so they can not deal with the runtime message evolution. Therefore, we need to extend the workflow engine so that it controls the data flow.

Storing and transmitting data in binary form is often desirable both to conserve I/O bandwidth and to reduce storage and processing requirements. PBIO (Portable Binary Input/Output) is a general approach to deal with binary data

in storage and transmission [6]. Users register the structure of the data that they wish to transmit/store or receive/read and PBIO transparently masks the differences [6]. We integrate the PBIO approach to support the messages exchange. It manages the input/output data by storing the streaming of data and matching the appropriate records with the relevant activity. Here, CAD's activity communicates with both Simulation and SR activities where the Simulation activity receives CAD's data as input and SR sends data to CAD as feedback.

5 Conclusions and Future Directions

In this paper, we have presented an approach for scientific workflow execution. Our approach exploits two distinct specifications for a scientific process: cooperative process and process messaging. We believe that anticipating activities and optimizing data streaming allow dynamic analysis and high performance communication. We plan to further implement this approach by developing a framework adapted for a scientific workflow execution based on Bonita and the PBIO approach.

Acknowledgements

This work was partially supported by Auraryd LLC. Software Company (France), within a bilateral cooperation with the ECOO research team (LORIA) on Scientific Workflow and Knowledge Infrastructure in Workflow Management.

References

1. Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping scientific workflows onto the grid. In *European Across Grids Conference*, pages 11–20, 2004.
2. Jia Yu and Rajkumar Buyya. A taxonomy of scientific workflow systems for grid computing. *SIGMOD Record*, 34(3):44–49, 2005.
3. Shawn Bowers and Bertram Ludäscher. An ontology-driven framework for data transformation in scientific workflows. In *DILS*, pages 1–16, 2004.
4. Claudia Bauzer Medeiros, Gottfried Vossen, and Mathias Weske. WASA: A workflow-based architecture to support scientific database applications (extended abstract). In *Database and Expert Systems Applications*, pages 574–583, 1995.
5. Daniela Grigori, François Charoy, and Claude Godart. Anticipation to enhance flexibility of workflow execution. In *DEXA*, pages 264–273, 2001.
6. Sandip Agarwala, Greg Eisenhauer, and Karsten Schwan. Lightweight morphing support for evolving middleware data exchanges in distributed applications. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 697–706, Washington, DC, USA, 2005. IEEE Computer Society.

Semantic Tools for Workflow Construction

Ondrej Habala, Marian Babik, Ladislav Hluchy,
Michal Laclavik, and Zoltan Balogh

Departement of Parallel and Distributed Computing, Institute of Informatics, Slovak
Academy of Sciences

Ondrej.Habala@savba.sk, Marian.Babik@saske.sk, hluchy.ui@savba.sk,
Michal.Laclavik@savba.sk, Zoltan.Balogh@savba.sk

Abstract. In this paper we present design and development of a knowledge framework for grid and web service-based workflow composition and execution. We highlight the corresponding architecture and the process of service annotation, discovery and composition in the project K-WfGrid [5]. We describe in detail the challenges of a flood-forecasting application and corresponding design and development of the service oriented model, which is based on the well known Web Service Resource Framework (WSRF). Semantic descriptions of the WSRF services are presented as well as the architecture, which exploits semantics in discovery and composition of service workflows. Further, we demonstrate how experience management solutions can aid the process of collaborative service discovery and composition. The whole K-Wf Grid system provides a unique approach in Semantic Grids by combining the advances of semantic web services and grid architectures.

1 Introduction

Recently, Web service (WS) technologies are gaining importance in the implementation of distributed systems, especially grids. One such example of WS implementation is the Web Service Resource Framework (WSRF) [3], which extends the current WS technologies by modeling the stateful services. Design and development of the service oriented distributed system is quite common and there are several emerging WS initiatives, which tries to automate the process of discovery, composition and invocation of services. The semantic web services are a typical example, showing the potential of how ontological modeling can improve the shortcomings of service oriented computing.

In this paper we will present the architecture and several interesting implementation details of a knowledge-based framework for workflow management in a service-based grid environment. The framework is based mainly on the knowledge of semantics of the environment, and the process of application workflow execution is driven by this knowledge. We therefore also briefly describe a corresponding architecture for discovery, composition and invocation of both stateful and stateless services and we provide a brief overview of the Web Ontology for Services (OWL-S). We show how a stateful service can be described in terms of

the OWL-S specification. Furthermore, we present the process of assisting the user in the composition and discovery of the services by using an experience management system based on text notes [13] as well as on instance based learning.

2 K-Wf Grid Architecture

Fig. 1 presents an architecture of the system components of the workflow orchestration and execution environment in the project K-Wf Grid. The main user interface for developing semantic-based Grid applications is the User Assistant Agent (UAA), which contacts the Grid Workflow Execution Service (GWES) that manages the process of composing and executing the services. The automated semantic service composition is partly delegated to the Automatic Application Builder (AAB), the Workflow Composition Tool (WCT), and the user (by means of the UAA). The Automatic Application Builder and the Workflow Composition Tool are knowledge-based semi-automatic modeling services, which in cooperation with the User Assistant Agent can propose known solutions to problems solved in the past. The semi-automatic composition of the services is enabled by the semantic description of the grid services, which is the main responsibility of the WSRF2OWL-S part of the Grid Organizational Memory (GOM). When parts of the workflow are ready to be executed on the Computing Grid, the Grid Workflow Execution Service asks the Scheduler for the optimal resource, due to some user-defined metrics. Then, the corresponding Web Service operation is invoked remotely on the Grid middleware using WSRF protocols. The events triggered by the workflow orchestration and execution will be published by means of an event system. The Knowledge Assimilation Agent (KAA) consumes these events and generates knowledge that is stored in the Grid Organizational Memory (GOM). This knowledge can be later reused by the components of the workflow orchestration and execution environment.

Workflow Composition Tool (WCT) provides the functionality of composing abstract workflows of Grid applications from simple user requirements. It employs semantic reasoning techniques over OWL-S descriptions (i.e. subsumption, classification) and it tries to propose a solution to the user's problem by using provided descriptions of available resources. Such a solution is delivered in form of an abstract workflow instance composed of service operations. This workflow of operations is based on the Petri nets model, which has several advantages over the directed acyclic graphs, for a detailed description see [7]. The main input to the WCT is a description of data (results) which is to be produced by the target application and, optionally, a set of user-provided data (input) to be used by the composed application. It is also possible to upload an incomplete workflow as an input in order to complete it automatically. The main output of the composition process is a refined description of the abstract workflow. During the composition and refinement of the workflow the User Assistant Agent is used to guide the user according to the experience it gained in the past compositions.

In the following we will describe more closely the WSRF2OWL-S translation, describing the semantics of the WSRF services, the actual implementation and

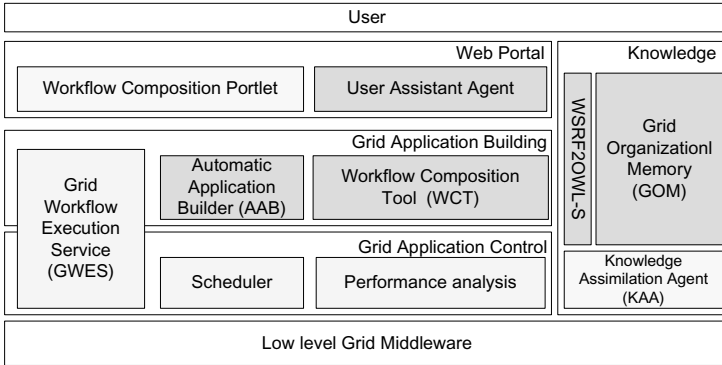


Fig. 1. A simplified scheme of K-Wf Grid architecture

use case of the system operation based on the flood-forecasting application. We will also describe the User Assistant Agent (UAA) and its role in the process of composition and refinement of the workflow. A detailed description of the Workflow Composition Tool and the Grid Workflow Execution Service can be found in [6, 7].

3 Adding Semantics to Stateful Services

3.1 Web Ontology for Services (OWL-S)

OWL-S is an ontology-based approach to the semantic web services [1]. The structure of the ontology consists of a service profile for advertising and discovering services, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with underlying service interfaces. Service profile (OWL-S profile) has functional and non-functional properties. The functional properties describe the inputs, outputs, preconditions and effects (IOPE) of the service. The non-functional properties describe the semi-structured information intended for human users, e.g. service name, service description, and service parameter. Service parameter incorporates further requirements on the service capabilities, e.g. security, quality-of-service, geographical scope, etc. Service grounding (OWL-S grounding) enables the execution of the concrete Web service by binding the abstract concepts of the OWL-S profile and process to concrete messages. Although different message specifications can be supported by OWL-S, the widely accepted Web Service Description Language (WSDL) is preferred [2].

3.2 Semantic Annotation of Services Based on OWL-S

Service annotation is the process of generating the semantic descriptions (i.e. OWL-S) of both stateless and stateful services from the web service descriptions (i.e. WSDLs). In K-Wf Grid it has become crucial in the process of providing

application support and enabling semantics for semantically unaware grid application areas. We have developed an annotation tool called WSRF2OWL-S. During its development we have faced several issues, mainly problems caused by the dynamic nature of resource properties of the WS-Resource standard and the complexity of stateful services and their interaction model.

3.3 WSRF2OWL-S Tool

We have designed and developed a tool for generating the OWL-S description for stateful and stateless services from the corresponding web service descriptions (WSDLs) [2]. Such tool is inevitable in the grid environment hosting a vast number of services, which have to be semantically described in order to enable automated discovery, composition and invocation.

The translation starts with a configuration and an URL of the WSDL document. The translator parses the WSDL document extracting the operations, port-types, inputs, outputs as well as resource properties. A combination of the WSDL4J [8], Axis WSDL [9] and Globus Toolkit WSDL utilities [4] are used in the process. The translator then generates for each WSDL operation a skeleton of the OWL-S document. Then it creates the inputs, outputs, preconditions and effects and maps the elements to the ontological concepts defined in the configuration. If needed, it will create an ontology, which models the resource properties of the given services. The GOMOWL-S API can be used to extend the OWL-S by the domain dependent constructs, e.g. FloodForecastingWSRFProfile, DataObjectInput, SimpleEffect, etc. The outcome of the process are OWL-S documents describing the web service operations, which are then be composed into the workflow as described in the next section.

4 Experience Management in the Discovery and Composition of Services

Knowledge and experience management [10, 11] is known more from area of information systems and organizational process management. However, we believe that such approach can be used also in the area of web service composition. When services are composed automatically, several composed workflows can be presented to the user. Based on available semantic description such workflows can be viewed as identical for the user problem. For example, if we compose services to predict weather forecast in Bratislava this can be fulfilled with MM5 or ALADIN meteorology service due to semantic description. However, one of the models may not give good results for certain geographical location or in certain season, and others can be more appropriate. Such knowledge can not be put into semantic description for all cases but can be easily described by expert users in form of text notes while the system is used. These notes (human experience) are processed using semantic annotation and the system detects semantic context (ontology concepts and individuals) of the note which is reviewed and confirmed by user. Such note with assigned context can thus be displayed in future to the user in similar context.

Use of semantic annotation is important for appropriate notes context detection and thus helping service composition when displaying relevant experience in actual user context. Annotation is also used for appropriate service discovery and to help user specify problem using free text which is translated to semantic description of the problem. Discovered semantic elements (user requirements) are then used by the system to compose services to fulfill the user problem. The main idea of the used annotation algorithm is to detect relevant structured knowledge described by a domain specific ontology model in the unstructured text. The main difference between existing annotation solutions such as Anotea [12] and EMBET is detection of ontology elements from existing domain ontology (other annotation solutions try to create such ontology).

The Knowledge Assimilation Agent (KAA) is used as another tool which enables knowledge to be exploited in Grid workflows and in Grid collaborative environments. The approach used in KAA deals with behavior prediction of Grid services. The performance measures of Grid services are estimated based on the past cases. Instance based learning (IBL) is used to estimate the performance of Grid services. We elaborate generic IBR methods by case retrieval refinement process through semantic description of discrete features and service input data. Case representation is crucial for capturing information about WS operation invocations. We model the case structure using ontology. In our approach a single case represents a single WS operation invocation (please see figure bellow).

Sample KAA scenario can be presented by the flood forecasting application (see Chapter 5) used in K-Wf Grid as a prototype demonstration. KAA is used to discover dependencies between execution time of a Grid service and the area for which the forecast is being computed. The geographic area is stored in a file (resource), which is semantically described. Ontology which describes the geographical area is used for refinement of case retrieval during performance prediction of the forecasting WS. Past grid service invocations give us information how long will a computation last for a given geographic area. Thus a user willing to submit a flood forecast for a certain area can estimate the length of forecast to be computed.

5 Service Based Flood-Forecasting Simulation Cascade

The flood prediction application (FFSC) is based on a a network of loosely coupled, cooperating but independent services. The K-Wf Grid's flood application became instantiated in several servers across the testbed. It is a set of loosely coupled services, with several possible execution scenarios. Main data provider and also external input to the cascade is the Slovak Hydrometeorological Institute (SHMI). The SHMI provides us with input data for the first stage of our application, the *Meteorology*. In this stage, we employ two models, *ALADIN* and *MM5*, for weather predictions, the latter having three distinct operation modes (simple, one-way nested and two-way nested). The predicted weather conditions are used in the *Watershed integration* stage to compute water runoff into the target river. This result is then further processed in the *Hydrology* stage, where two models - *HSPF* and *NLC* - compute river levels for selected geographical

points. These levels are then used to model water flow in the last, *Hydraulic* stage of the application. Concurrently all important results are optionally visualized, packaged and displayed to the user - if required.

Apart from the simulation models, preprocessor and associated tools, the data flow contains also several *job packagers* and a *User Proxy Service*. These services implement our approach towards interactive grid jobs and also toward *in-process user collaboration*. The user proxy service may receive a ZIP file (prepared by a visualization service) with a HTML sub-tree, which is displayed to a certain user. The user is then notified of his/her new task (currently by e-mail, later also ICQ notifications will be implemented) and may view the HTML sub-tree, included images, animations, tables, etc. and may even react and provide input by filling out HTML forms (which are further processed by following services in the application workflow). For each specialised task another user may be asked to provide input, thus collaborating on a bigger job, requiring expertise of several users. Apart from enabling seamless multi-user collaboration on a single workflow, the concept of the User Proxy Service enables (with its asynchronous notifications) users to leave long-running workflows unattended and return to them on request, either to view the computed results or to provide some input.

5.1 Example Application Scenario

When a user logs into the system, he/she starts by using the User Assistant Agent interfaces (see Fig. 2). He/she enters a text description of the problem to be computed. This description is then analyzed for known keywords and detected elements are presented to the user to confirm detected context of the problem. Free text problem definition is important, when a user starts to work with the system and wants to define the problem in a way understandable for automatic service composition.

When some elements of the problem context are confirmed by the user, they become the semantic representation of user's problem and composition of the services can start. Detected ontology elements are from the service ontology generated by the WSRF2OWL-S as well as from other domain ontologies related to the input and output data, such as geographical ontology of the target area (in case of geographically bound simulations, for example).

After the problem context and semantic description is stated, the system creates a so-called abstract workflow, consisting only of unknown input, one high-level, abstract task (transition) and the defined output (the solution). This abstract workflow is then analyzed, relevant services capable of creating the solution are found, and a more concrete workflow is created, in which the abstract transition is replaced by several classes of services. These are not yet actual service instances and calls, but rather representations of certain capabilities, which are known to be found in one or more real service instances (see the light gray boxes in Fig. 2).

At this point, the system is ready to start execution of the workflow. When the user clicks the play button in the workflow visualization (lower left part of Fig. 2), the system starts looking for real service instances, which are able to perform the

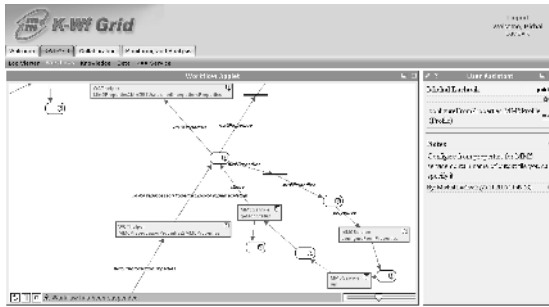


Fig. 2. K-Wf Grid portal with User Assistant pane on the right, workflow visualization on the left

tasks represented by the class transitions. These service instances are evaluated by the KAA based on their previous monitored behaviour, and the instance believed to perform best (according to a selected metric, for example speed) is then executed. If the system is unable to find service instance for the class transition, user's attention is required. Also, the system is able to recover from a fault of the selected service instance, and to use another instance, possibly working one.

The User Assistant pane has another important role in the workflow execution process. To aid user in service selection, input data provision and general orientation in the world of web and grid services, it provides description of the workflow elements, if such description (note) has been entered previously by other user. This is yet another form of experience management, this time based on the text notes passed between users. These text notes are entered in a form of a line (or several lines) of text through a button in the User Assistant pane. After the note is entered, the context of the currently selected element of the workflow is analysed, verified by the user and the note is bound to this semantic context. Then the User Assistant is able to find the note later, if a similar context of a workflow element is present in the workflow, and the note may be displayed and may possibly guide the user with previous user's experience. These notes may be used to describe certain special qualities (or deficiencies) of some service classes or instances, such as ability/inability to work under certain conditions or to provide quality results for some tasks.

6 Conclusion

The presented K-Wf Grid system is primarily intended to simplify composition and execution of service-based workflows. It also offers several useful collaboration capabilities, which make it very usable in teams composed of several domain experts and application users. The system is able to learn from previous successes and failures. It can be used for a wide range of applications, from long-running simulations to interactive grid applications, requiring user input at several places

of their workflow. The applications can be easily extended and the system is able to use both stateless web services as well as the new stateful WSRF-based grid services.

Acknowledgments. The research reported in this paper has been partially financed by the EU within the project IST-2004-511385 K-WfGrid and Slovak national projects, Research and development of a knowledge based system to support workflow management in organizations with administrative processes, APVT-51-024604; Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources, SPVV 1025/04.

References

1. A. Ankolekar et.al, OWL-S: Semantic Markup for Web Service, 2003, <http://www.daml.org/services/owl-s/1.1>
2. E. Christensen, F. Cubera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, Technical report, WWW Consortium, 2001, <http://www.w3c.org/TR/wsdl>
3. Web Service Resource Framework, <http://www.globus.org/wsrfl/>
4. Globus Toolkit, <http://www-unix.globus.org/toolkit/>
5. The Knowledge-based Workflow System for Grid Applications FP6 IST project. <http://www.kwfgird.net>
6. Gubala, T., Bubak, M., Malawski, M., Rycerz, K., Semantic-based Grid Workflow Composition, In: Proc. of 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, R.Wyrzykowski et.al. eds., 2005, Springer-Verlag, Poznan, Poland
7. Hoheisel, A., User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2005
8. IBM WSDL4J Project, <http://oss.software.ibm.com/developerworks/projects/wsdl4j>
9. Apache WebServices - Axis Project, <http://ws.apache.org/axis/>
10. Thomas H. Davenport, Laurence Prusak, Working Knowledge, ISBN 1578513014, May, 2000
11. Ralph Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, ISBN 3540441913, 2002
12. Annotea Project, <http://www.w3.org/2001/Annotea/>, (2001)
13. Laclavik, M., Gatial, E., Balogh, Z., Habala, O., NGuyen, G., Hluchy, L., Experience Management Based on Text Notes (EMBET), eChallenges Conference, 19 - 21 October 2005, Ljubljana, Slovenia ,2005

Stochastic Modeling and Quality Evaluation of Workflow Systems Based on QWF-Nets*

Yunni Xia, Hanpin Wang, Chunxiang Xu, and Liang Li

School of Electronics Engineering and Computer Science
Peking University, China
{xiayunni, whpxhy}@pku.edu.cn

Abstract. Quality (QOS) prediction is one of the most important research topics of workflow management system. In this paper, we propose the SWQ approach to analytically evaluate QOS of workflow systems based on QWF-net, which extends traditional WF-net by associating tasks with exponential response time and exponential TTF (time-to-failure). The comparison between simulative and analytical results in the case study indicates that the SWQ approach achieves satisfactory accuracy. The paper concludes with a comparison between the SWQ approach and other related work.

1 Introduction

With the advent and evolution of global scale economies, organizations need to be more competitive, efficient and flexible. In the past decade, workflow techniques have been widely used to address these needs.

However, among many research topics of workflow, performance/quality analysis is yet to be given the importance it deserves. Techniques and models [1-8] for QOS evaluation are still preliminary and limited. This paper introduces an analytical approach, the SWQ (meaning a stochastic approach for quality evaluation of workflow systems) to address the need for QOS evaluation. The SWQ approach is based on QWF-net (stochastic WF-net) model, which is an extension of traditional WF-net where tasks are associated with exponential response time and exponential TTF (time-to-failure). By mapping the execution of QWF-net into a homogeneous continuous Markovian process, the SWQ approach calculates its expected completion-time and reliability.

Through comparing simulative results and results obtained by SWQ approach, the case-study indicates that our approach achieves satisfactory accuracy. The paper concludes with a comparison between the SWQ approach and related work.

* Supported by the National Natural Science Foundation of China under Grant No. 60173002 and the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312004.

2 QWF-Net for Quality Prediction

The Workflow net (WF-net) proposed by van der Aalst is a high level Petri Nets with two special places i and o , which indicate the beginning and the end of the modeled process. There exist four routing patterns in WF-nets, namely sequence, parallel, selective and iterative. WF-net does not care the concept of time, but sometimes we need to consider time aspect in workflow management systems. For example, we want to know the completion time of the whole workflow net or of some subnets so that we can decide whether the arrangement of the workflow system meets constraint of time. So introducing time concept into WF-net is necessary.

This section extends WF-net to QWF-net by associating an exponential response time and exponential TTF (Time-to-failure) with each task.

Definition 1. (QWF-net) $N = (P, T, Task, \lambda, \mu)$ is a QWF-net if and only if:

1. N is structurally a WF-net
2. SPLIT/JOIN transitions (black thin bars in Fig.1) fire immediately and have response time of 0
3. SPLIT/JOIN transitions never fail
4. The set $Task \subseteq T$ denotes the set of transitions excluding SPLIT/JOIN transitions (as illustrated by white bars in Fig.1)
5. Each task has an exponential response time. A function $\lambda : Task \rightarrow Real$ is used to identify parameter (also known as **execution-rate**) of the exponential response time of each task
6. Each task fails independently and has an exponential TTF. A function $\mu : Task \rightarrow Real$ is used to identify parameter (also known as **failure-rate**) of the exponential TTF of each task

It easily follows that QWF-net is identical with WF-net in construction aspect. Therefore, structural properties of WF-net also follow in QWF-net: there should be no dead tasks; the procedure should terminate eventually; at the moment the procedure terminates there should be one token in sink place o and all the other places are empty; the definition of reachable markings and its corresponding calculation methods for WF-net can also be applied to QWF-net.

3 Quality Evaluation Based on QWF-Net

Besides assumptions about exponential task response time and TTF, this paper also assumes that:

1. The control flow randomly chooses its path on **XOR-split** according to given choice probabilities. For generality, this paper uses a function $se : Task \rightarrow Real$ to denote the probability that each task is selected when its preceding task finishes execution. Note that, if a task is not on any XOR-split, its choice probability equals 1, otherwise smaller than 1

- The control flow skips **loop** when current iteration finishes according to a given probability. For generality, this paper uses a function $lo : Task \rightarrow Real$ to denote the probability that the control flow skips each task when current iteration finishes. Note that, if a task is not on any loop, its corresponding skipping probability equals 1, otherwise smaller than 1. Therefore, for a task t_i , its expected number of loop iterations is $\frac{1}{lo(t_i)}$, following a **geometric distribution**.

3.1 Evaluating the Expected Completion Time (ECT)

This subsection presents an analytical method to calculate the expected completion time of QWF-net based on response time of each task. Let $U(t)$ denote the set of operational tasks in QWF-net at time t (execution begins at time 0), then its state-space (denoted by S) is obtained through mapping each reachable marking into a corresponding set of operational tasks.

For any reachable marking $M_1 \in [M_0 >$ where no SPLIT/JOIN transitions are activated, there exists a state which records all operational tasks in this marking. Since SPLIT/JOIN transitions fire immediately with response time 0, there is no need to generate states indicating whether SPLIT/JOIN transitions are activated or not. Take Fig.1 for example, the marking illustrated in this figure where P_1 contains a token is one such negligible marking since this marking merely indicates that $AND - split_1$ transition is activated. Also note that the marking where only sink place contains a token is mapped into an **absorbing state** which records no task is operational, meaning all tasks are idle and the control flow terminates.

The state space of $U(t)$ of Fig.1 are illustrated in Table.1. Note that, there exist more than one **initial-state** since $XOR - SPLIT_1$ may generate one token into place P_4 , P_5 or P_6 . S_{12} is the absorbing state.

Table 1. State space

state	operational tasks	state	operational tasks
S_1 (Initial-state)	$\{t_1, t_6\}$	S_7	$\{t_3, t_8\}$
S_2 (Initial-state)	$\{t_1, t_7\}$	S_8	$\{t_3\}$
S_3 (Initial-state)	$\{t_1, t_8\}$	S_9	$\{t_6\}$
S_4	$\{t_1\}$	S_{10}	$\{t_7\}$
S_5	$\{t_3, t_6\}$	S_{11}	$\{t_8\}$
S_6	$\{t_3, t_7\}$	S_{12} (Absorbing-state)	\emptyset

As mentioned in the assumption, a task t_i has exponential response time with parameter $\lambda(t_i)$ and the probability that control flow skips task t_i when current iteration of t_i finishes is $lo(t_i)$ ($lo(t_i) = 1$ if t_i is not on loop). The number of loop iterations of t_i , N_{t_i} , is a geometric distributed random variable with parameter $lo(t_i)$. Since task may be iteratively executed, this paper uses D_{t_i} to denote the total response time of t_i considering iterative execution and X_{t_i} to denote

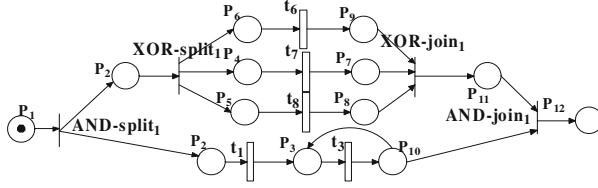


Fig. 1. A QWF-net example (Case5)

response time of one single iteration. According to the definition of QWF-net, X_{t_i} follows exponential distribution. Therefore, the cumulative-distribution-function (CDF) of D_{t_i} is given as

$$\begin{aligned}
 F(y) &= Prob\{D_{t_i} \leq y\} \\
 &= \sum_{K=1}^{\infty} Prob\{N_{t_i} = k\} Prob\{N_{t_i} \times X_{t_i} \leq y | N_{t_i} = K\} \\
 &= \sum_{K=1}^{\infty} lo(t_i)(1 - lo(t_i))^{K-1} E_K(y)
 \end{aligned} \tag{1}$$

where $E_K(y)$ denotes the CDF of **K-phase Erlang distribution**.

Then, the density-distribution-function (PDF) of D_{t_i} is given as

$$\begin{aligned}
 f(y) &= F'(y) \\
 &= \sum_{K=1}^{\infty} lo(t_i)(1 - lo(t_i))^{K-1} \frac{\lambda(t_i)(y\lambda(t_i))^{K-1}}{(K-1)!} e^{-\lambda(t_i)y} \\
 &= \lambda(t_i)lo(t_i)e^{-y\lambda(t_i)} \sum_{K=1}^{\infty} \frac{((1 - lo(t_i))y\lambda(t_i))^{K-1}}{(K-1)!} \\
 &= \lambda(t_i)lo(t_i)e^{-y\lambda(t_i)} \times e^{(1-lo(t_i))\lambda(t_i)y} \\
 &= \lambda(t_i)lo(t_i)e^{-\lambda(t_i)lo(t_i)y}
 \end{aligned} \tag{2}$$

where $\frac{\lambda(t_i)(y\lambda(t_i))^{K-1}}{(K-1)!} e^{-\lambda(t_i)y}$ is the PDF of the K-phase Erlang distribution.

According to the equation above, D_{t_i} follows exponential distribution with parameter $\lambda(t_i)lo(t_i)$. Since the total response time D_{t_i} of every task t_i follows exponential distribution, $U(t)$ is a homogeneous continuous Markovian process. The infinitesimal generator matrix Q of $U(t)$ is given as

$$q_{i,j} = \begin{cases} lo(t_i) \times \lambda(t_i) \times \prod_{t_m \in NEW(i,j)} se(t_m) & \text{if } S_i \xrightarrow{t_i} S_j \\ -\sum_{1 \leq r \leq W, r \neq i} q_{i,r} & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{3}$$

where $lo(t_i) \times \lambda(t_i)$ denotes the parameter of the exponential random variable D_{t_i} , W denotes the number of states in the state space, and $q_{i,j}$ denotes the transition rate from state S_i to S_j .

Relation $S_i \xrightarrow{t_l} S_j$ means that S_j is the resulting state of S_i if the operational task t_l in S_i finishes execution and becomes idle. Note that, there may exist more than one resulting states of S_i when t_l becomes idle because transition may activate choice (XOR-split). Those resulting states are viewed as different **types** in the Markovian chain according to the **phase-type** property and $\prod_{t_m \in NEW(i,j)} se(t_m)$ denotes the occurrence probability of S_j among all types, where **NEW**(i,j) denotes the set of newly-emerging operational tasks in the transition from state S_i to S_j .

Take the transition from state S_8 to S_{12} for example, we have its transition rate $q_{8,12}$ as $lo(t_3)\lambda(t_3)$ according to Eq.3.

The expected time which a state S_i takes to reach the absorbing state (time-to-termination) is defined as ETT_{S_i} , where

$$ETT_{S_i} = \begin{cases} 0 & \text{absorbing-state} \\ \frac{1}{E_{S_i}} + \sum_{1 \leq k \leq W, k \neq i} \frac{q_{i,k} \times ETT_{S_k}}{E_{S_i}} & \text{else} \end{cases} \quad (4)$$

where $E_{S_i} = \sum_{1 \leq j \leq W, j \neq i} q_{i,j}$

According to the equation above, the expected time for S_i to reach termination is the expected duration of state S_i ($\frac{1}{E_{S_i}}$) plus the averaged (by occurrence probabilities) time-to-termination of its immediate succeeding states.

Therefore, the expected completion-time of QWF-net is the averaged (by occurrence probability) time-to-termination of all initial states. Let *Init* and $BUSY_{S_i}$ denote the set of initial states and the set of operational tasks in state S_i respectively, then the QWF-net’s expected completion time *ECT* is

$$ECT = \sum_{S_i \in Init} (ETT_{S_i} \times \prod_{t_j \in BUSY_{S_i}} se(t_j)) \quad (5)$$

3.2 Evaluating Reliability

The reliability estimate of t_i is obtained through integrating the probability that its TTF is larger than t (meaning the probability that t_i does not fail till time t) multiplied by PDF of D_{t_i} over the interval from 0 to ∞ . Therefore, reliability of task t_i , R_{t_i} , is given by

$$\begin{aligned} R_{t_i} &= \int_0^\infty \lambda(t_i)lo(t_i)e^{-\lambda(t_i)lo(t_i)t} \times Prob\{TTF_{t_i} > t\} dt \\ &= \int_0^\infty \lambda(t_i)lo(t_i)e^{-\lambda(t_i)lo(t_i)t} \times e^{-\mu(t_i)t} dt \\ &= \frac{\lambda(t_i)lo(t_i) \int_0^\infty (\lambda(t_i)lo(t_i) + \mu(t_i))e^{-(\lambda(t_i)lo(t_i) + \mu(t_i))t} dt}{\lambda(t_i)lo(t_i) + \mu(t_i)} \\ &= \frac{\lambda(t_i)lo(t_i)}{\lambda(t_i)lo(t_i) + \mu(t_i)} \end{aligned} \quad (6)$$

where $\lambda(t_i)lo(t_i)e^{-\lambda(t_i)lo(t_i)t}$ is PDF of the D_{t_i} and $e^{-\mu(t_i)t}$ is the probability that TTF (time-to-failure) of t_i is greater than t .

Therefore, QWF-net’s reliability is the averaged reliability of all initial states

$$Reliability = \sum_{S_i \in Init} (R_{S_i} \times \prod_{t_j \in BUSY_{S_i}} se(t_j)) \tag{7}$$

where R_{S_i} denotes the reliability of state S_i

$$R_{S_i} = \begin{cases} 1 & \text{absorbing-state} \\ \sum_{Every S_j \text{ satisfying } S_i \xrightarrow{t_l} S_j} R_{t_l} \times R_{S_j} \times \frac{q_{ij}}{E_{S_i}} & \text{else} \end{cases} \tag{8}$$

where $E_{S_i} = \sum_{1 \leq j \leq W, j \neq i} q_{i,j}$

4 Case Study and Simulation

This section applies the SWQ approach to some examples and studies its accuracy through a comparison with Monte-Carlo simulation. The examples are given by Fig.2. Tasks involved are listed in Table.2. $Case_{1-4}$ are four simple examples dealing with sequential, parallel, selective and iterative routing modes, respectively. $Case_5$ (given earlier in Fig.1) and $Case_6$ are more complex examples featured by all routing modes.

A simulation procedure is developed to evaluate expected completion time and reliability of QWF-net. The simulation procedure is similar to simulation

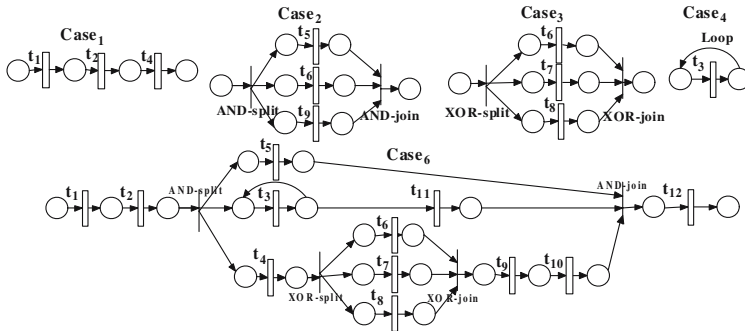


Fig. 2. Cases

Table 2. Tasks involved in the cases

Tasks	λ	μ	se	lo	Tasks	λ	μ	se	lo
t_1	0.3	0.0026	1	1	t_7	0.3	0.0013	0.2	1
t_2	0.2	0.0028	1	1	t_8	0.45	0.0014	0.1	1
t_3	0.4	0.0053	1	0.33	t_9	0.3	0.0064	1	1
t_4	0.65	0.0041	1	1	t_{10}	0.45	0.0054	1	1
t_5	0.6	0.0051	1	1	t_{11}	0.8	0.0023	1	1
t_6	0.2	0.0037	0.7	1	t_{12}	0.25	0.0058	1	1

algorithms for SPN (stochastic Petri-net) performance analysis, which use random variable generators to decide firing delay of each transition at each simulation run. The procedure is executed for sufficiently many times. At each run, the procedure randomly chooses a path along XOR-split and decides the numbers of loop iterations of loops according to predefined probabilities. Then the procedure uses exponential-distribution random variable generators to generate response time of each task. Also, the program uses random variable generators to generate the TTF of every task. At each run, if all tasks' TTF are greater than their response time (meaning no failure happens during the execution of this task), a success is recorded. The simulative estimates of completion-time is obtained through averaging completion-time of all runs. The simulative estimate of reliability is the ratio of successes to the number of all runs.

Results obtained by simulation (illustrated in normal style) are compared with those by the SWQ approach (illustrated in bold style) in Table.3. As shown, analytical results is pretty close to simulative results. It indicates that SWQ approach achieves satisfactory accuracy.

Table 3. Comparison between simulative and SWQ results

case	ECT	Reliability	case	ECT	Reliability
<i>Case₁</i>	9.8879/ 9.8718	97.25%/ 97.16%	<i>Case₄</i>	7.4942/ 7.5000	96.07%/ 96.13%
<i>Case₂</i>	6.5456/ 6.5480	95.42%/ 95.32%	<i>Case₅</i>	11.8686/ 11.8661	93.94%/ 93.99%
<i>Case₃</i>	4.3883/ 4.3889	98.60%/ 98.61%	<i>Case₆</i>	25.5926/ 25.5539	86.07%/ 86.11%

5 Comparison with Related Work

Research of [2,3] uses a reduction technique to simplify sequential, parallel, selective and iterative routing patterns into a single transition with equivalent QOS estimate, however these methods are not very realistic since they assume tasks in WF-net have deterministic response time rather than nondeterministic as this paper does.

Methods of [1,7,8] are similar to the SWQ approach in that they also model the control flow as continuous Markov chains. These models map execution of each task into each state of CTMC chain and analytically evaluate transition probability or transition rate between states. However these methods can not model parallel execution of more than one tasks in their Markov chains.

Method proposed by [5] develops a simplification technique to simplify four basic routing patterns into a single task with approximate equivalent performance and assumes that the **equivalent task** still follows exponential distribution if all tasks have exponential response time. However, its assumption that the simplified construct still follows exponential distribution is obviously inaccurate and unrealistic. For instance, *n* tasks with exponential response time of the same execution rate arranged by sequential routing pattern should be simplified into a single equivalent task of *n*-phase Erlang response time (which is obviously not exponential).

6 Conclusion

This paper proposes the SWQ method to analytically evaluate QOS (expected-completion-time and reliability) of workflow systems based on the QWF-net model, which extends traditional WF-net by associating tasks with exponential response time and TTF. This paper also develops a simulation procedure to calculate the simulative QOS results. The comparison between simulative and analytical results in the case study indicates that SWQ approach achieves satisfactory accuracy.

References

1. J.Klingemann, J.Waesch and K.Aberer. Deriving Service Models in Cross-OrganizationalWorkflows. In *Proceedings of Int'l Workshop on Reasearch Issues in Data Engineering (RIDE)*, pages 100-107, 1999
2. Jorge Cardoso, Amit Sheth, John Miller, Jonathan Arnold and Krys Kochut. Quality of service for workflows and web service processes. In *Elsevier Transaction on web semantics*, 1(3), pages 281-308, 2004
3. Zhangxi Tan, Chuang Lin, Hao Yin, Ye Hong and Guangxi Zhu. Approximate Performance Analysis of Web Services Flow Using Stochastic Petri Net. In *Proceedings of Grid and Cooperative Computing GCC 2004 Third International Conference*, pages 193-200, 2004
4. Michael C.Jaeger, Gregor Rojec-Goldmann and Gero Muhl. QoS Aggregation in Web Service Compositions. In *Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service EEE-05*, pages 181-185, 2005
5. LIN Chuang, QU Yang, REN Fengyuan and Dan C.Marinescu. Performance Equivalent Analysis of Workflow Systems Based on Stochastic Petri Net Models. In *Proceedings of 1st International Conference on Engineering and Deployment of Cooperative Information Systems*, pages 64-79, 2002
6. JianQiang Li, YuShun Fan, and MengChu Zhou. Performance Modeling and Analysis of Workflow. In *IEEE transaction on SYSTEMS, MAN, AND CYBERNETIC-SPART A: SYSTEMS AND HUMANS*, 34(2), pages 229-242, 2004
7. Michael Gillmann, Jeanine Weissenfels, Gerhard Weikum and Achim Kraiss. Performance and Availability Assessment for the Configuration of Distributed Workflow Management Systems. In *Proceedings of 7th International Conference on Extending Database Technology*, pages 183-201, 2000
8. Michael Gillmann, Gerhard Weikum and Wolfgang Wonner. Workflow Management with Service Quality Guarantees. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 228-239, 2002

Styx Grid Services: Lightweight, Easy-to-Use Middleware for Scientific Workflows

J.D. Blower¹, A.B. Harrison², and K. Haines¹

¹ Reading e-Science Centre, Environmental Systems Science Centre,
University of Reading, Reading RG6 6AL, UK
jdb@mail.nerc-essc.ac.uk

² School of Computer Science, Cardiff University, Cardiff CF24 3AA, UK

Abstract. The service-oriented approach to performing distributed scientific research is potentially very powerful but is not yet widely used in many scientific fields. This is partly due to the technical difficulties involved in creating services and composing them into workflows. We present the Styx Grid Service, a simple system that wraps command-line programs and allows them to be run over the Internet exactly as if they were local programs. Styx Grid Services are very easy to create and use and can be composed into powerful workflows with simple shell scripts or more sophisticated graphical tools. Data can be streamed directly from service to service and progress can be monitored asynchronously using a mechanism that places very few demands on firewalls. Styx Grid Services can interoperate with Web Services and WS-Resources.

1 Introduction

The concept of “workflow” in e-Science terminology refers to the composition of high level modules (which are often distributed, Internet-based services such as Web Services) in order to create an application. For example, a scientist may wish to extract data from a number of data archives in different physical locations, perform some analysis on these data on a high-performance resource in another location, then produce some visualization of the end result on his or her local machine. The services in this workflow are mutually independent (“loosely coupled”) and may be hosted by a number of different service providers.

In theory, this approach should allow scientists with little technical knowledge to create powerful distributed applications. In practice, however, there are – at the time of writing – very few examples of scientific communities that have started to work in this way on a routine basis. A large part of the reason for this is the paucity of services that are available for scientists to use.

Web Services provide very significant advantages for creating loosely-coupled, interoperable services: they are accessed through XML messaging and are thus inherently cross-platform, they are self-describing (through Web Service Definition Language – WSDL – documents) and are a widely-accepted standard for distributed computing. However, Web Services have some important limitations in the context of scientific workflows. In particular, it is impractical to

encode anything but a trivial amount of data in XML due to the processing time required and the inflating effect of doing so. Furthermore scientific services are often long-running and so it is highly desirable to be able to monitor the progress and status of the service as it runs using asynchronous notifications. Solutions such as OGSF [1] and the Web Services Resource Framework (WSRF, <http://www.globus.org/wsrp/>) employ notification mechanisms that require the client to run a server process. This requirement means that clients that are behind stringent firewalls or Network Address Translation (NAT) systems will not receive these notifications.

If scientists are to adopt the workflow approach in their work, there must exist a set of useful services from which these workflows can be constructed. In order to achieve such a “critical mass” of services, it must be possible for *scientists* to be able to create such services with minimal or no help from dedicated technical staff. Several systems exist to make the task of creating Web and Grid Services easier (e.g. Soaplab [2] and GEMICA [3]). However, these systems are still typically difficult for scientists to use, either because the scientists are not familiar with the Web or Grid Services model or because the systems are based on complex, heavyweight toolkits such as Globus (<http://www.globus.org/>), which are designed for application builders, not end users. Therefore, technical support is needed to create these services and the critical mass of useful services is never reached. Once created, it is important that the services be as easy as possible to use.

There is a clear demand from scientists [4] for simple, lightweight middleware that does not necessarily support every possible feature but that is easy to install, use and understand. This demand has resulted in the recent development of systems such as WEDS [5].

We have developed a solution that addresses all of the above issues. We focus on the process of creating services that are based on command-line programs (which may be tried-and-tested “legacy” codes) but the principles we describe could be extended to other service types. The solution we present deliberately moves away from the Web Services model but, as we shall demonstrate, still maintains a high level of interoperability.

We introduce the Styx Grid Services (SGS) system, a framework for wrapping command-line (i.e. non-graphical) programs and allowing them to be run as a service from anywhere on the Internet. The major advantages are:

- It is very easy to create SGSs that wrap command-line programs.
- Remote SGSs can be used *exactly* as if they were locally-installed programs.
- Workflows can be created using simple shell scripts or graphical tools.
- Data can be streamed directly between remote service instances.
- The software is very lightweight and quick to install (less than 5 MB, including all dependencies).
- The software places few demands on firewalls, requiring only one incoming port to be open on the server and *no* incoming ports to be open on client machines.

2 Styx Grid Services: Background

Our main goal in developing the Styx Grid Services system was to create remote services that are just as easy to use as local programs. The basis of the system is the Styx protocol for distributed systems [6]. In Styx-based systems *all* resources are represented as files, analogous to the representation of the mouse as the file `/dev/mouse` in Unix variants. Styx is a file-sharing protocol that can operate over a large number of transports such as TCP/IP and UDP. It forms the core of the Inferno and Plan9 operating systems, in which applications communicate with all resources using Styx, without knowing whether these resources are local or remote (in Plan9, Styx is known as “9P”). We developed an open source, pure-Java implementation of Styx (JStyx, <http://jstyx.sf.net>) and used it as the base for the SGS system.

All resources in Styx systems are represented as a file hierarchy, which is known as a *namespace*. We have defined a namespace that represents a command-line program [7]. Clients interact with this program by reading from and writing to the files in this namespace over the network. For example, the SGS namespace contains an `inputs/` directory, into which clients write the input files that the program will consume.

Due to this filesystem-like structure, every resource on a Styx server can be represented very naturally as a URL. For example, the file that represents the standard output stream of instance 1 of the the `mySGS` service can be represented by the URL `styx://<server>:<port>/mySGS/instances/1/outputs/stdout`. This is very important in the context of workflows: these URLs are passed between services in a workflow to enable direct transfer of data between services (see Sect. 4.1).

The Styx protocol itself deliberately does not mandate any particular security mechanism. In JStyx, we secure systems using transport-layer security (TLS), using public key certificate-based authentication and (optional) encryption of network traffic. This encryption is transparent to applications that use JStyx.

When Styx clients and servers interact they typically use *persistent connections*: the client connects to the server and leaves the connection open for as long as it needs. This means that the client can receive asynchronous messages from the server without requiring any incoming ports to be open through its firewall. Also, the client does not need a public IP address so it does not matter if the client is behind a NAT router. This is how we solve the problem of asynchronous notification that was discussed in Sect. 1 above. A single Styx server can handle multiple tasks (such as asynchronous messaging and file transfers) and so servers only need to have a single incoming port open through the firewall. This helps to make the deployment and use of Styx systems very easy.

3 Wrapping Programs as Styx Grid Services

Neither service providers nor end-users need to know anything about the technical details discussed in Sect. 2 above. The process of wrapping a command-line

program as a Styx Grid Service is very simple. A short XML description of the program in question is constructed. This description is a complete specification of the program, specifying the command-line parameters and input files that the program expects and the output files that the program produces. (There is other optional information that can be added, but that is beyond the scope of this paper.) A server program is then run that parses the XML file and sets up the SGS namespace. A single server can host many Styx Grid Services. Note that the executable itself cannot be read over the Styx interface.

3.1 Executing SGSs Just Like Local Programs

Once the program is deployed as a Styx Grid Service, it can be run from anywhere on the Internet, *exactly as if it were a local program*. For example, consider a program called `calc_mean` that reads a set of input files (perhaps from a set of scientific experiments), calculates their mean and writes the result to an output file. If this service were deployed on the server `remotehost.com`, listening on port 9092, and the user has a set of input files (called `input1.dat`, `input2.dat` etc.) the user would run the service by entering the following command:

```
SGSRun remotehost.com 9092 calc_mean input*.dat -o mean.dat
```

The `SGSRun` program is a general-purpose command-line client for any Styx Grid Service and it performs the following tasks: It connects to the server and downloads the XML description of the Styx Grid Service that it is being asked to run. It uses this description to parse the command-line arguments that the user has provided. If these are valid, it creates a new instance of the service and sets its parameters, based on these command-line arguments. It then uploads the necessary input files, starts the service running and downloads the output data as soon as they are produced. If the SGS uses the standard streams (stdout, stderr and stdin) these are redirected to and from the console as appropriate.

It is an easy task to create a simple wrapper script called `calc_mean` on the client. This wraps the `SGSRun` program and contains the location and port of the remote server. Then this wrapper script can then be treated *exactly* as if it were the `calc_mean` program itself.

4 Creating Workflows from Styx Grid Services

4.1 Using Shell Scripts as Workflows

Given that remote SGSs can be executed exactly like local programs, workflows can be created with simple shell scripts. Workflows are simply high-level programs and so it is natural to use a scripting environment to create them. This allows SGSs to be combined easily with local programs and permits the use of all the programming features that the scripting language provides (e.g. loops and conditionals). Let us consider a simple workflow of two Styx Grid Services. The first is the `calc_mean` service from the above example. The second SGS,

called `plot`, takes a single input file and turns it into a graph. The shell script (workflow) that would be used to take a set of input files, calculate their mean and plot a graph of the result would be:

```
calc_mean input*.dat -o mean.dat
plot -i mean.dat -o graph.gif
```

(1)

Note that this is *exactly the same script* as would be used to invoke the programs if they were installed locally. (This assumes that the user has created wrapper scripts called `calc_mean` and `plot` that invoke the `SGSRun` program as described above.)

Direct data passing. The above “workflow” (shell script) is very simple but not optimally efficient. The intermediate file `mean.dat` is not required by the user: it is simply uploaded to the `plot` service as soon as it is downloaded. This wastes time and bandwidth. The intermediate file can be *passed directly between the services* with only a minor change to the script:

```
calc_mean input*.dat -o mean.dat.sgsref
plot -i mean.dat.sgsref -o graph.gif
```

(2)

The `.sgsref` extension is a signal to the system to download a *reference* (URL) to the output file and place it in the file `mean.dat.sgsref`. This reference is then passed to the `plot` service, which downloads the real file directly from the `calc_mean` service. Hence this intermediate file does not pass through the workflow enactor (i.e. the client’s machine). See Fig. 1.

Data streaming using the pipe operator. Let us imagine that the `calc_mean` program outputs data to its standard output, instead of writing to an output

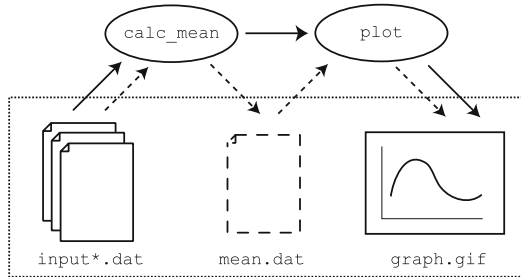


Fig. 1. Illustration of direct data passing between Styx Grid Services. The ellipses are Styx Grid Services and the dotted box represents the client’s machine. The dashed arrows represent data transfers that result from the workflow in script 1 in section 4.1. The intermediate file `mean.dat` is not required by the client and so the workflow can be arranged (script 2 in section 4.1) so that this file is passed directly between the SGSs (solid black arrows).

file. Similarly, imagine that the `plot` program reads data from its standard input and outputs the picture to its standard output. The command required to execute the workflow (with both local programs *and* Styx Grid Services) is:

```
calc_mean input*.dat | plot > graph.gif (3)
```

Here, the intermediate data are being streamed to the local client, then streamed back out to the `plot` service. We can ensure that the intermediate data are streamed directly between the services with a minor change to the command:

```
calc_mean input*.dat --sgs-ref-stdout | plot > graph.gif (4)
```

The `--sgs-ref-stdout` flag is a signal to send a reference (URL) to the standard output of the `calc_mean` service to the standard input of the `plot` service. In this way the intermediate data are streamed directly between the services, across the Internet.

Weaknesses of this approach. The inputs and outputs of SGSs are files and so the “workflow engine” (i.e. the shell environment) performs no type checking on these entities. The responsibility of checking for validity of inputs is left to the services themselves. Secondly, although constructs such as loops are supported by the shell, the variables used to control these loops cannot be read directly from the outputs of SGSs. An important subject of future research would be to use the SGS approach to wrap entities such as classes and functions, rather than whole executables: in this case, inputs and outputs could be strongly typed and could also be captured by workflow engines, solving the above two problems.

4.2 Using Graphical Workflow Tools

The command line scripting interface to the SGS system that is described above is perhaps the simplest way of creating SGS workflows. In some cases, however, there are significant advantages in using more sophisticated graphical tools to interact with services and create workflows. In particular, graphical interfaces can provide richer interactivity with the SGS server: progress and status can be monitored graphically and the service can be steered [7].

The Taverna workbench (<http://taverna.sf.net>) is a graphical workflow system that was designed for performing *in silico* experiments in the field of bioinformatics, but it is sufficiently general to be useful to other communities. We have worked with the Taverna developers to incorporate support for Styx Grid Services into the software. Using Taverna, the user can build workflows by mixing diverse service types, including Web Services and SGSs.

The Triana workflow system (<http://trianacode.org>) is a graphical workflow environment that can interface with many different service types (including Web Services), but cannot currently interface directly with Styx Grid Services. We have developed two ways for achieving this:

1. **Brokering:** A separate Web Service is created that accepts SOAP messages and uses the information therein to communicate with an SGS server [7].
2. **“SOAP over Styx”:** The Styx Grid Service itself is modified to accept SOAP messages that are written directly to a special file in its namespace using Styx. The SGS describes itself using a WSDL document that is also readable via a special file. This WSDL document defines service operations that encapsulate the messages and data to be written to the files in the SGS namespace. So for example, to tell the SGS to read its input data from a certain URL, the client invokes the `setStdin(String url)` operation that is defined in the WSDL. We have built support for this into WSPeer [8], the Peer-to-Peer oriented Web Service framework that is used by Triana.

4.3 Wrapping SGSs as WS-Resources

The Web Services Resource Framework (WSRF) is a recent specification which addresses the need to handle resources that maintain state across service invocations. “WS-Resources” are resources that are exposed and manipulated via a Web Service. A Styx Grid Service is exposed as a WS-Resource by transforming its configuration information (Sect. 3) into *ResourceProperties*, which are QName/value pairs of a specified data type that are used to describe a WS-Resource in WSDL. SGSs define certain properties which map directly onto WSRF specifications. For example, the `time/` directory in the SGS namespace, which houses files containing data pertinent to the lifetime of the service, can be mapped onto the properties defined in the WS-ResourceLifetime [9] specification. The `serviceData/` directory of the SGS namespace contains state data which clients can subscribe to and receive notifications of changes from. These are exposed as WS-Notification [10] topics.

WSPeer is capable of wrapping an SGS as a WS-Resource in two ways. The first way (brokering) involves creating a WSRF service that receives SOAP messages over HTTP and translates the information therein into Styx messages, which it sends to a separate SGS server. The second is to use the Styx protocol itself to send and receive XML, as described in Section 4.2. The ability of WSPeer to use the Styx protocol directly allows clients that are behind firewalls and NAT systems to receive WS-Notification messages via the notification mechanism described in Sect. 2. While it is useful to expose SGS functionality according standard specifications, we do not attempt to wrap the SGS data streams in XML for performance reasons. For example an output stream exposed as a *ResourceProperty* consists of a URI, while the actual data in the stream is application specific.

5 Conclusions

We have introduced a new type of Internet service, the Styx Grid Service (SGS). SGSs wrap command-line programs and allow them to be run from anywhere on the Internet, exactly as if they were local programs. SGSs can be combined

into workflows using simple shell scripts or more sophisticated graphical workflow engines. Data can be streamed directly between SGS instances, allowing workflows to be maximally efficient. We have shown that Styx Grid Services can operate as part of a Web Services or WSRF system through the use of methods including broker services.

A key strength of the SGS system is that it is very easy to create and use services: it is well within the reach of most end-users (scientists) to do so with no help from dedicated technical staff. Problems connected with firewalls and NAT routers are vastly reduced compared with other systems, allowing for easy deployment and use. We believe that the Styx Grid Services system represents a significant step forward in increasing the usability of service-oriented systems and workflows in science.

Acknowledgements

The authors would like to thank Tom Oinn for incorporating the SGS framework into Taverna and Vita Nuova Holdings Ltd. for technical help with the Styx protocol. This work was supported by EPSRC and NERC, grant ref. GR/S27160/1.

References

1. Tuecke, S., et al.: Open Grid Service Infrastructure (OGSI) Version 1.0. Technical Report GFD-R-P.15, Global Grid Forum (2003)
2. Senger, M., Rice, P., Oinn, T.: Soaplab - a unified Sesame door to analysis tools. In Cox, S., ed.: Proceedings of the UK e-Science Meeting. (2003) ISBN 1-904425-11-9.
3. Kacsuk, P., Kiss, T., Goyeneche, A., Delaitre, T., Farkas, Z., Boczkó, T.: A high-level grid application environment to Grid-enable legacy code. *ERCIM News* **59** (2004)
4. Chin, J., Coveney, P.V.: Towards tractable toolkits for the Grid: a plea for lightweight, usable middleware. UK e-Science Technical Report UKeS-2004-01, http://www.nesc.ac.uk/technical_papers/UKeS-2004-01.pdf (2004)
5. Coveney, P.V., Vicary, J., Chin, J., Harvey, M.: WEDS: a Web services-based environment for distributed simulation. *Phil. Trans. R. Soc. A* **363** (2005) 1807–1816
6. Pike, R., Ritchie, D.M.: The Styx architecture for distributed systems. Online, <http://www.vitanuova.com/inferno/papers/styx.html> (1999)
7. Blower, J., Haines, K., Llewellyn, E.: Data streaming, workflow and firewall-friendly Grid Services with Styx. In Cox, S., Walker, D., eds.: Proceedings of the UK e-Science Meeting. (2005) ISBN 1-904425-53-4.
8. Harrison, A., Taylor, I.: WSPeer - An interface to Web Service hosting and invocation. In: HIPS-HPGC Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models. (2005)
9. OASIS: Web Services Resource Lifetime 1.2 (WS-ResourceLifetime). Online, http://docs.oasis-open.org/wsrif/wsrif-ws_resource_lifetime-1.2-spec-pr-02.pdf (2005) Public Draft 02.
10. OASIS: Web Services Base Notification 1.2 (WS-BaseNotification). Online, <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf> (2004) Draft 03.

Automatic Services Composition in the Grid Environments*

Wenju Zhang, Fei Liu, Shudong Chen, and Fanyuan Ma

Shanghai Jiaotong University, Shanghai,
P.R. China, 200030

{zwj03, liufei001, chenshudong, fyma}@sjtu.edu.cn

Abstract. Different planning techniques have been proposed so far which address the problem of automated composition of web services. However, in realistic cases, the planning problem is far from trivial. Such issue is more serious when services are embraced in grid environments, which are usually resource-conscious. We propose a planning techniques for the automated composition of grid services described in OWL-S process models. The technique allows for the synthesis of plans that encode compositions of grid services with the usual programming constructs. We apply this technique in our DDGrid project and do some preliminary experimental evaluations.

1 Introduction

Currently, the problem of the composition of services is addressed by two orthogonal efforts from the industry and the Semantic Web community. Our research aims to develop intelligent middleware components that encapsulate the expertise required to use grids. Earlier, we used AI planning techniques to automatically generate executable job workflows from high-level specifications of desired results. We integrated our planner in a grid environment to extract relevant knowledge from existing grid middleware.

The rest of this paper is organized as follows. Section 2 introduces the grid environments of our techniques apply. Section 3 presents how OWL-S process models can be translated into state transition systems and the principle and definition of automatic grid services composition. Section 4 provides an example and a preliminary experimental evaluation. Section 5 discusses the relevant ontologies and planning techniques. We conclude in Section 6.

2 Grid Environments and Integration

Middleware services such as those that the Globus toolkit provides help users obtain information about available resources, component software, data files and

* This research work was supported in part by the the National High Technology Research and Development Program of China (863 Program), under Grant No. 2004AA104270.

the execution environment. We have used several of these services as knowledge sources for our planner as part of its integration in the DDGrid (Drug Discovery Grid, www.ddgrid.ac.cn) project for planning and execution in grids. An AI planner and the typical services composition environment shown in Figure 1.

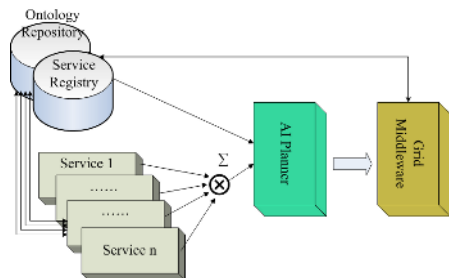


Fig. 1. An AI planner and the typical services composition environment

A complete planning-based solution for Grid or Web services composition must address execution as well as construction of workflows. Grid and Web planning systems make decisions in dynamic environments in which the services that are composed as part of a plan may become unavailable during its execution.

Finding an optimal allocation of processors for tasks in a workflow is NP-hard and tools must focus on finding reasonable heuristics or on identifying families of problems that can be solved efficiently. In a dynamic environment, the system needs to make time-dependent decisions when an initial workflow allocation is found to be inappropriate during execution.

3 OWL-S Process Models and Automated Composition

OWL-S process models are declarative descriptions of the properties of web service programs. Process models distinguish between atomic processed and composite processes as mentioned before.

We encode OWL-S process models as state transition systems, which describe dynamic systems that can be in one of their possible states and can evolve to new states as a result of performing some actions.

The execution structure corresponding to domain Σ and plan π is the Kripke structure $\Sigma_\pi = \langle S, S_0, R \rangle$, where:

- S is the set of configurations;
- $S_0 \subseteq P$ are the initial configurations;
- $R \subseteq S \times S$ are the transitions between configurations.

The execution structure Σ_π represents the evolutions of the domain Σ controlled by the plan π . It is the execution structure Σ_π that must satisfy the composition goal G . If $\Sigma_\pi \models G$, we say that π is a valid plan for G on Σ .

In presence of partial observability, at each plan execution step, the plan executor has to consider a set of domain states, each equally plausible given the initial knowledge and the observed behavior of the domain so far. Thus in general, given a belief B , performing an action a and taking into account the obtained observation o gets to a new belief $Evolve(B, a, o)$:

$$Evolve(B, a, o) = \{s' : \exists s \in B. s' \in \mathcal{T}(s, a) \wedge \mathcal{M}(s') = o\}.$$

Planning in this framework consists in searching through the possible evolutions of initial beliefs, to retrieve a conditional course of actions that leads to beliefs that satisfy the goal.

4 An Example and Experimental Evaluation

Our techniques are general, but we have applied them in the context of DDGrid project. The Drug Discovery Service (as shown in Figure 2), also called Composed Grid Service (CGS), is composed by three related grid services we provided: a Virtual Chemical DataBase Data Service, a Dock Service and a Payment Service. The idea is that of combining these three services so that user may directly request the CGS to complete the molecule docking task.

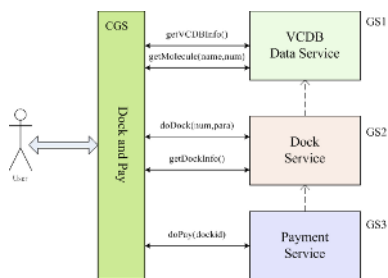


Fig. 2. A DDGrid Services Composition Example

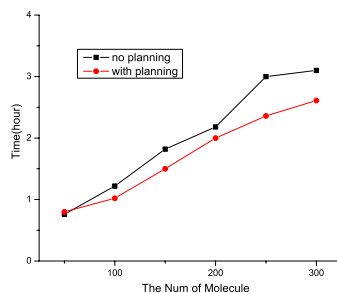


Fig. 3. The Execution Time of the CGS with Different Number of Molecule

The Initial State: The planner receives as input an initial state that captures information from several sources, including

- Hardware/Software resources available to the user. those information obtain from the grid middleware of DDGrid.
- The dock application's specifical parameters. such as the protein file description, the ligand atom file description, etc.
- Estimates of computation power of every Virtual Organization and bandwidths between the resources.
- The identification of the grid user and the authorization policy.

Composition Goals: The composition goals express requirements for the services to be automatically generated. In our example (see Figure 2), The CGS should try to reach the ideal situation when the user has submitted his job, and the VCDB data service has the corresponding molecule to dock. Finally, the dock grid service finish the job and the user pay for it through the payment service.

We have executed the composed grid service in our DDGrid platform. The planner performed 6 actions, docking 1050 molecules. The effectiveness and performance can be seen partially from Figure 3.

5 Related Work

Several projects have recently demonstrated planning techniques for Web and Grid Services. Other projects use knowledge bases to facilitate Grid use. The myGrid project uses OWL and OWL-S to describe application components as Semantic Web Services. These descriptions support matching and discovery of components through a description logic reasoner.

Other planning techniques have been applied to related problems in the field of grid and web services [1, 3, 4]. The interactive composition of information gathering services has been tackled in [2] by using Constraint Satisfaction Problems techniques.

6 Conclusion and Future Work

In this paper, we have shown how ontologies and OWL-S process models can be used to generate automatically new composed grid services. This is achieved by translating OWL-S process models to state transition systems and by generating automatically a plan that can express conditional and iterative behaviors of the composition. Our preliminary experimental evaluation shows the potentialities of the approach.

References

1. J. Blythe, E. Deelman, and Y. Gil, Automatically Composed workflows for Grid Environments, IEEE Intelligent Systems, 2004.
2. S. Thakkar, C. Knoblock, and J.L. Ambite. A View Integration Approach to Dynamic Composition of Web Services, In Proceedings of ICAPS'03 Workshop on Planning for Web Services, 2003.
3. William K. Cheung et al., Towards Autonomous Service Composition in A Grid Environment, In Proceedings of the IEEE International Conference on Web Services(ICWS'04), 2004.
4. Tatiana Kechkaylo, Planning for Grid Applications with Explicit Reservations, The 14Th International Conference on Automated Planning & Scheduling(ICAPS'04), 2004.

A Non-intrusive and Incremental Approach to Enabling Direct Communications in RPC-Based Grid Programming Systems

Alexey Lastovetsky, Xin Zuo, and Peng Zhao

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland
{Alexey.Lastovetsky, Xin.Zuo, Peng.Zhao}@ucd.ie

Abstract. This paper advocates a non-intrusive and incremental approach to enabling existing Grid programming systems with new features. In particular, it presents a software component enabling NetSolve applications with direct communications between remote tasks. The software component is a supplementary one working on the top of the basic NetSolve system. Its design also allows remote tasks to be freely mixed in a single application, independent on whether each particular task is enabled for direct communications or not. Experiments with this software are also presented.

1 Introduction

High performance Grid programming systems have reached a certain level of maturity. Two examples are NetSolve [1-3] and Ninf [4] that allow scientific programmers to develop reliable Grid applications. On the other hand, the constantly growing number of users and applications results in the need of further development of such systems in terms of functionality and quality.

Traditionally, addition of a new feature to a Grid programming system is achieved by changing the code of the system to produce its new version. This approach to the evolution of Grid programming systems has two serious disadvantages. First of all, the change of the system's code may introduce bugs and result in some applications not running properly anymore or even crashing. Secondly, the new version of the system has to replace the old version on all computers of the Grid in order to support the development and execution of applications enabled with the new feature. Such simultaneous and total replacement can have very high organizational overhead and sometimes be simply unrealistic as different computers on the Grid are managed and administered by independent and, very often, loosely connected users.

The goal of our research is to investigate if an existing Grid programming system can be enabled with new features in a non-intrusive and incremental way. *Non-intrusiveness* means that the original system does not change and the new features are provided by a supplementary software component working on the top of the system. *Increment* means that the supplementary software component does not have to be

installed on all computers to enable applications with the new features. It can be done step by step and the new features will be enabled in part, with the completeness dependent on how many nodes participating in the execution of the application have been upgraded with the supplementary software component. In this paper, we use NetSolve and one particular feature that is direct communication between remote tasks, to demonstrate the feasibility of the non-intrusive and incremental evolution of Grid programming systems.

The rest of the paper is structured as follows. Section 2 describes in detail the design and implementation of a supplementary software component enabling NetSolve applications with direct communications between remote tasks in a non-intrusive and incremental way. Section 3 presents some experiments with this software. Section 4 outlines related work and concludes the paper.

2 Enabling Direct Communications in NetSolve

NetSolve is positioned as a programming system for high performance distributed computing on global networks based on GridRPC [5]. In NetSolve, output data of remote tasks are typically sent back to the client upon completion of each remote task even if the data are only needed as input for some other remote tasks, resulting in so-called bridge communications which increase the execution time of applications. In this paper, we propose a lightweight supplementary software component that enables direct communication between remote tasks in NetSolve in a non-intrusive and incremental way, without recompilation or reinstallation of the original NetSolve programming system. We start presenting the software component by a short description of its use. The only thing for **client programmers** to do is to install the wrapper API and Job Name Service on the client side, then compile the client program with the wrapper library. The **procedure developers** should do nothing to enable direct communications and develop their own procedures as usual. To enable direct communications on server side, **the server administrator** needs to register the software component as a new problem file to NetSolve. No re-installation and re-compilation for the system. The proposed software component consists of three parts: Client API & Argument Parser, Server Connector and Job Name Service (JNS).

Client API provides a uniform interface for the client to make remote procedure calls. Despite the modification on the remote side, the wrapper API allows the calls to be made in the same manner. The only difference is in the arguments that can be not only variables storing real data but also handlers, which is a variable storing real data, the local IP address and the port number are used as such communication info. If input argument is a handler, then a request is sent to the JNS to get the IP address and the port number of the remote resource and this information is used as communication info for this handler. If this output argument is a handler, the returned result information from computational servers is sent to JNS and registered there. In this sense, upon making a call to NetSolve, only a handler array that is transferred to the remote server. The Server Connector manages all the other I/O data transaction.

Server Connector is on the server side, which is a proxy program responsible for interacting with clients and other Server Connectors to enable direct communications. When all necessary data have been acquired by Server Connector, it re-submit to the

local host to perform computations that the user exactly requested for. There is no difference in the way the client and computational servers download the result of the computations. The Server Connector firstly returns the result's communication information to the client. Then it sets up a socket waiting for the client or the server to connect in to download the result of computations.

Job Name Service (JNS) is responsible for registration of procedure upon its invocation during RPC call. Other procedures may send requests to the JNS to search for registered procedure. During the execution of the application, it contains all information about every handler. Only client has the permission to register or access a handler on the JNS. There is no communication and interaction between JNS and computational servers. Because JNS is designed as a system-independent system on the client side, it can be applied to different RPC-based systems and not influenced by any fault or crash on the server side.

3 Implementation and Experiments

For our experiments we choose the same remote computational task that has been used in experiments with REDGRID presented in [6], namely, matrix multiplication. Experiments in [6] used 2 remote servers to perform 3 matrix multiplications, and the client, agent and servers all were in the same Ethernet segment. In our experiments, we used 8 remote servers to perform 8 matrix multiplications. The interconnecting network is based on 100 Mbit Ethernet with a switch enabling parallel communications between computers.

Based on our experiments results we can make conclusion that communication cost is visibly reduced by using direct communications, where seven communication bridges were eliminated among twenty four communications. So, the theoretical speedup is $7/24 = 29.2\%$. The obtained experimental speedup ranges from 24% to 27%, which is close to the theoretical value. We can also see that the experimental results are similar to the REDGRID ones, which are ranging from 18% to 28%. The speedup depends on the ratio of the number of eliminated bridge communications and the total number of communications. If communication links connecting remote computers are much faster than communication links connecting the remote computers and the client computer, the speedup will be much higher. Another experimental results show that speedup is around 54% while bridge communications is performed at the rate of 10 Mbit/sec and the direct communications between remote servers is performed at the rate of 100 Mbit/sec.

4 Related Works and Conclusion

To enable direct communications, NetSolve introduces an original mechanism called Request Sequencing [7]. The most restrictive of which is that all the tasks have to be performed on the same computing node. Another effort to reduce the overhead of bridge communications in NetSolve is the Logistical Computing and Internetworking (LoCI) [8]. The mechanism is mainly aimed at replicating data in order to keep them even in the case of crash of some of the computers. Although it is sufficient for

enabling direct communications, the goal of building a complete network storage system makes LoCI over-heavy for enabling just this particular feature. The REDGRID project [6] is closest to our approach sharing the similar idea behind its design, which uses an intrusive and non-incremental approach and requires re-compilation and re-installation of the modified NetSolve on all involved computing nodes. The main difference is that REDGRID is built into NetSolve and difficult to be migrated to other GridRPC-based systems. Another related project is SmartNetSolve [9], an extension of NetSolve aimed at higher performance of Grid applications, which also enables direct communications in an intrusive and non-incremental way.

In this paper, we have presented an approach to reducing unnecessary bridge communications in RPC-based Grid programming systems. The main advantage of the approach is that it is non-intrusive, requiring no changes in the enabled programming system. It does *NOT* need recompilation or reinstallation of the Grid programming system. The approach is incremental by nature allowing remote tasks both enabled for direct communication and not, to be freely mixed in a single application. It can be applied to different RPC-based Grid programming systems. Finally the experimental results have shown that the performance of Grid applications can be significantly improved by using our supplementary software component.

This work was supported by the Science Foundation Ireland.

References

1. <http://icl.cs.utk.edu/netsolve/>
2. Casanova H., Dongarra J.: NetSolve: A Network Server for Solving Computational Science Problems. The International Journal of Supercomputer Applications and High Performance Computing, Vol. 11, No. 3, pp. 212--223, 1997
3. Arnold D., Casanova H., Dongarra J.: Innovation of the NetSolve Grid Computing System. Concurrency: Practice and Experience, Vol. 14, No. 13-15, pp. 1457-1479, 2002
4. Tanaka, Y., Nakada, H., Sekiguchi, S., Suzumura, T., Matsuoka, S.: Ninf-G: A reference implementation of RPC-based programming middleware for Grid computing. Journal of Grid Computing, Vol.1, No.1, pp. 41--51, 2003
5. Seymour, K., Nakada, H., Matsuoka, S., Dongarra, J., Lee, C., Casanova, H.: Overview of GridRPC: A Remote Procedure Call API for Grid Computing. In: Proceedings of the Third International Workshop on Grid Computing, pp. 274--278, Springer-Verlag, 2002
6. Desprez, F., Jeannot, E.: Improving the gridrpc model with data persistence and redistribution. In: Proceedings of ISPDC 2004 / HeteroPar'04, pp. 193--200, IEEE Computer Society, 2004
7. Arnold, D., Agrawal, S., Blackford, S., Dongarra, J., Miller, M., Seymour, K., Sagi, K., Shi, Z., Vadhiyar, S.: Users' Guide to NetSolve V1.4.1. Technical Report ICL-UT-02-05, University of Tennessee, Knoxville, TN, 2002
8. Beck, M., Arnold, D., Bassi, A., Berman, F., Casanova, H., Dongarra, J., Moore, T., Obertelli, G., Plank, J., Swamy, M., Vadhiyar, S., Wolski, R.: Middleware for the use of storage in communication. Parallel Computing, Vol. 28, No. 12, 2002
9. Brady T., Konstantinov E., Lastovetsky A.: SmartNetSolve: High Level Programming System for High Performance Grid Computing. In: Proceedings of IPDPS 2006, IEEE Computer Society, 2006

Enacting Proactive Workflows Engine in e-Science

Ezio Bartocci, Flavio Corradini, and Emanuela Merelli

Università di Camerino, Dipartimento di Matematica e Informatica
Camerino, 62032, Italy
{ezio.bartocci, flavio.corradini, emanuela.merelli}@unicam.it

Abstract. The dynamic nature and the geographic distribution of scientific resources, require flexible and adaptive computational environment where an in-silico experiment can be executed as a workflow of activities. In this paper, we propose a software environment to dynamically generate domain-specific, agent-based workflow engines from workflow specifications. The workflow engine is a proactive multiagent system -a distributed, concurrent system- whose autonomous components interact in performing the workflow activities in a specific domain. The proposed approach has been implemented on Hermes, agent-based mobile computing middleware, and tested within “Oncology over Internet” project.

1 Introduction

Over the past few years, new high-throughput methods for data collection in life science, e.g. microarray processing, have greatly increased data generation, and the wide use of the Web has fostered scientists to share data sources and computational methods in a collaborative way. As consequence, the traditional scientific process has become computationally intensive and *in-silico* experiments -described as processes of concurrent activities- are laboriously executed in a large, distributed and dynamic environment.

Nowadays, e-Science -the use of advanced computing technologies to support scientist- seems to be the only way to face the complexity of the scientific computational environment. We believe that workflow and agent-based technologies [2] together with an effective and efficient resource management system [1] could be a good start to face the complexity that surrounds scientist’s work.

An in-silico experiment can be naturally specified as a workflow of activities implementing the data analysis process in a standardized environments. The workflow owns the advantage to be reproducible, traceable and compositional by reusing other workflows; features that are fundamental to validate a scientific experiment. The software component that “defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic”, according to Workflow

Management Coalition (WfMC) Reference Model [5], is named Workflow Management System (WMS). In e-Science domain, although several systems that support the daily work of a bioscientist have been proposed in literature [7, 8, 4], they are not compliant to the WfMC Reference Model. And, none of them have been designed to face unforeseen circumstances and to take on the fly decisions. Moreover, workflows are generally static and workflow engines centralize the execution and the coordination of the computation.

In this work, we intend to overcome the above limitations by proposing the dynamic generation of a workflow engine, associated to a single workflow specification. Our approach exploits the proactiveness and mobility of agent-based technology to embed the application domain features inside the agents behavior. The resulting workflow engine is a multiagent system -a distributed, concurrent system- typically open, flexible, and adaptive.

In the remainder of this paper we describe our workflow engine architecture based on agent technology, then, we explain our implementation on an agent-based mobile middleware. The approach is illustrated with a case study on the "Oncology over Internet" project.

2 Agent-Based Workflow Engine

A workflow is a distributed application that involves the coordinated execution of human and system activities, usually, in an heterogeneous environment. Based on our previous work [2], we consider a workflow as coordination model for a pool of agents -workflow executors- that implements the workflow engine for a specific workflow instance. In this context agents are autonomous active entities, encapsulate the execution of independent activities, execute their tasks, concurrently to the work of the other agents, cooperate, in their autonomy, for a common goal forms a multiagent system (MAS) [6].

In our framework, a user, after defining a workflow with a graphical notation, compiles the workflow specification in a set of primitives performing the behaviours of a pool of agents. This approach has been implemented on Hermes [3] architecture. Hermes is an agent-based mobile middleware. The choice to use Hermes has been conditioned by the 3-layers -user, system, runtime-, component-based architecture that facilitates the management of domain-specific components [2] toward the development of a workflow-to-MAS compiler. Moreover, Hermes supports the agent mobility through code mobility. User layer allows designers to specify their application as a workflow of activities using the graphical notation. System layer provides a domain-specific context-aware compiler to generate a pool of user mobile agents from the workflow specification. Run-time layer supports the activation of a set of domain-specific service agents and it supports agent discovery, mobility, creation, communication and security. Service agents in the run-time layer are localized in one platform to interface the local execution environment. User agents in the system layer are workflow executors. They are dynamically created for a specific goal that, in theory, is reached in a finite time by interacting with other agents.

3 Case Study: Oncology over Internet Project

The proposed approach has been tested within the Oncology over Internet (O2I) project (<http://www.o2i.it>). The main goal of the project has been the design of a framework to support searching, retrieving and filtering information from Internet in the oncology research and clinics domain. The O2I system architecture is shown in Figure 1. It includes three main components: the workflow manager (WCA), the user interface (UI) and the workflow executor (WE). Workflows are created and tested by administrator both in XSculf using Taverna Workbench and in XPDL -a WfMC standard- using JaWE -an open source workflow editor that supports XPDL. The user interface supports end users authentication and profiling and allows for the selection and launch of workflows (see Figure 2).

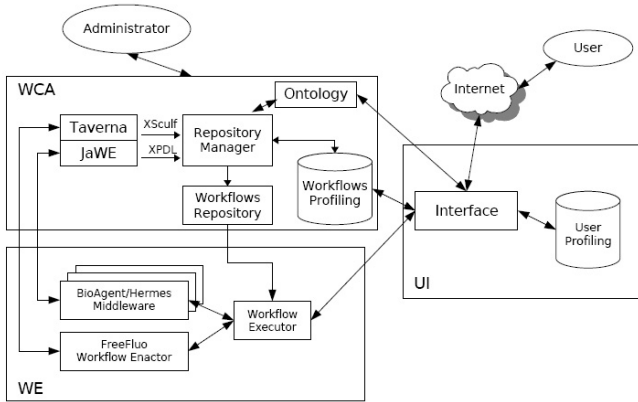


Fig. 1. The general O2I system architecture includes three main blocks: workflow creation and annotation (WCA), user interface (UI) and workflows execution (WE)

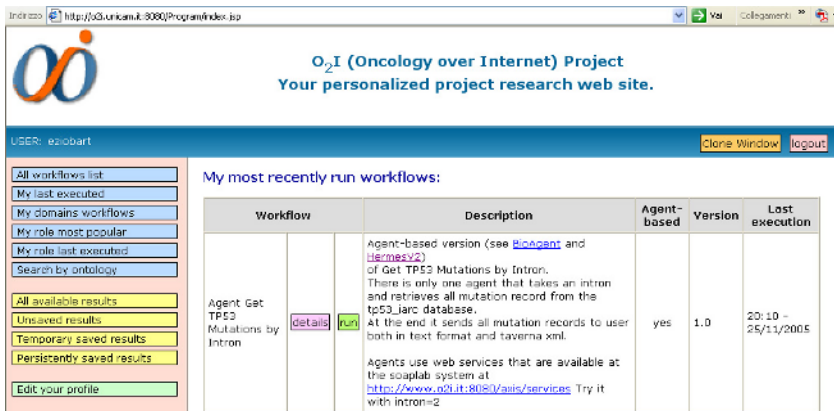


Fig. 2. Screenshot of O2I Portal

Workflows are executed either on FreeFluo Workflow Enactor by a centralized execution or on Hermes by a proactive, decentralize execution of BioAgents. While the first is used to carry out the Taverna workflow execution, the second compiles XPD L specification into a pool of mobile workflow executors. In the latter case the workflow execution is carried out by proactive, cooperative bio-user agents that interact with bio-service agents through messages exchange, and when necessary, decentralize the workflow execution exploiting mobility.

4 Conclusion

In this work, we proposed an agent-based Workflow Engine that exploits the proactiveness of agents to adapt to a dynamic, domain-specific, execution environment. We developed a prototype for the biomedical domain; the resulting Workflow Management System, according to the Workflow Management Coalition Reference Model, supports the dynamic generation of proactive workflow engines.

Acknowledgements

This work was partially supported by the Italian Ministry of Education, University and Research (MIUR), projects “Oncology over Internet (O2I)” and “Laboratory of Interdisciplinary Technologies in Bioinformatics (LITBIO)”.

References

1. N. Cannata, E. Merelli, and R. B. Altman. Time to organize the bioinformatics resourceome. *PLoS Comput Biol.*, 1(7):e76, 2005.
2. F. Corradini, L. Mariani, and E. Merelli. An agent-based approach to tool integration. *Journal of Software Tools Technology Transfer*, 6(3):231–244, 2004.
3. F. Corradini and E. Merelli. Hermes: agent-base middleware for mobile computing. In *Mobile Computing*, volume 3465, pages 234–270. LNCS, 2005.
4. A. Garcia Castro, S. Thoraval, L. Garcia, and R. MA. Workflows in bioinformatics: meta-analysis and prototype implementation of a workflow generator. *BMC Bioinformatics*, 6(1):87, 2005.
5. D. Hollingsworth. The Workflow Reference Model, January 1995.
6. N. R. Jennings. On agent based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000.
7. T. Oinn et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54, 2004.
8. S. Shah et al. Pegasys: software for executing and integrating analyses of biological sequences. *Bioinformatics*, 1(5):40, 2004.

Traffic Noise and Maximum-Flow Spanning Trees on Growing and Static Networks

Bosiljka Tadić¹ and Stefan Thurner²

¹ Department for Theoretical Physics, Jožef Stefan Institute,
Box 3000, 1001-Ljubljana, Slovenia

² Complex Systems Research Group, HNO, Medical University of Vienna,
Währinger Gürtel 18-20, A-1090 Vienna, Austria

Abstract. Properties of traffic noise and flow are often measured on complex networks and are used to diagnose the network's functional state and underlying structure, even though the precise structure–function interdependences are often unknown. Here we attempt to unravel some basic interdependences between structure and traffic on networks in numerically controlled traffic models. We simulate constant-density traffic on two different network topologies, which emerge from the same preferential rewiring rules but one within growth and the other under static conditions. We determine universal noise properties and the maximal-flow spanning trees on these classes of network topologies. We study both low-density traffic (structure dependences) and high-density traffic, where queuing influences transport properties.

1 Introduction

Dependences of network topology and optimal network functioning are gaining increased attention in network research lately [1, 2, 3]. While networks driving e.g. biological processes have adapted and optimized their structures through evolutionary mechanisms, whose details are largely unknown [4], artificial networks, on the other hand, often suffer from all kinds of dis-functional problems and weaknesses. Both theoretical and practical aspects of the *structure–function* interdependences are of interest.

In real networks, measurements of dynamic variables, such as flow through individual links, fluctuations of noise at individual nodes, round-trip times of information packets, high-throughput measurements etc., are often used to diagnose the network's functional state and structure. The idea is that the temporal fluctuations of the local node and link activities in an intricate way reflect the underlying network structure and potential dynamic correlations. The nature of these interdependences, however, is often hidden in the non-linearity of the dynamics and in most cases not known. In the absence of precise theoretical grounds, often largely simplified formal procedures are applied for these purposes, often severely missing crucial aspects of the studied systems. It is therefore of great importance to identify the essential elements of the structure–function interdependences in complex networks in a systematic manner. One approach is

to use numerical models of network dynamics in which a controlled variation of both structural parameters and the dynamic rules can be done systematically and inter-correlations can be detected and quantified. Moreover, robustness can be checked against varied dynamic conditions.

Here we apply an approach of this kind to study information transport on networks within a numerical model with *driving, search, and queuing* which we introduced in Refs. [5, 6, 7]. We use two classes of networks which are grown from the same *preferential rewiring* rules and same control parameters, but with different conditions that are known as evolving and static networks. By implementing the traffic rules on these networks, we show how they perform the information transport within the same dynamical conditions of a stationary traffic flow. Within the model we measure the travel times of packets, flow at all links on the network and temporal fluctuations of all node activities. In the low-density limit of our dynamics when packet density $\rho \rightarrow 1$, representing the infinitely slow driving (or sequential packet dynamics), these measures incorporate most of the network structural effects on the transport. In the high density limit we measure waiting times of packets at individual nodes and global noise correlations, which give quantitative estimates of the queuing effects on networks. In the low-density limit we present quantitative results for the flow intensity along network links, which defines the maximum-flow spanning tree of the network, as well as noise correlations and queue distributions, when the packet density is kept high. We show how the observed properties of traffic are related to the structural characteristics of both types of networks.

2 Structures and Traffic Rules

2.1 Network Structures

We consider two types of networks grown from *preferential attachment* [8] and *rewiring* rules [9], in which we apply different constraints known as evolving and static conditions. (i) In the evolving case, the network is grown by sequentially added nodes from $i = 1, 2, \dots, N$, with the preferential attachment and preferential rewiring rules according to the time dependent probabilities p_{in} and p_{out} . They are applied in the subset of *pre-existing* nodes at each growth step i . The linking probabilities depend on the current number of incoming q_{in} and outgoing q_{out} links at a node [9]

$$p_{in}(k, i) = \frac{\alpha + q_{in}(k, i)/M}{(1 + \alpha)i} \quad p_{out}(n, i) = \frac{\alpha + q_{out}(n, i)/M}{(1 + \alpha)i} \quad . \quad (1)$$

These rules lead to power-law distributions of both incoming and outgoing links [9], a large clustering coefficient and link correlations [10], a structure which is statistically very similar to the actual WWW. This is why this network is called *Web graph* (WG). An example of the emergent structure is shown in the left column of Fig. 1. (ii) For the static graph we apply the same rules as in Eq. (1), however, the links are added sequentially for the fixed number of nodes $i = N$.

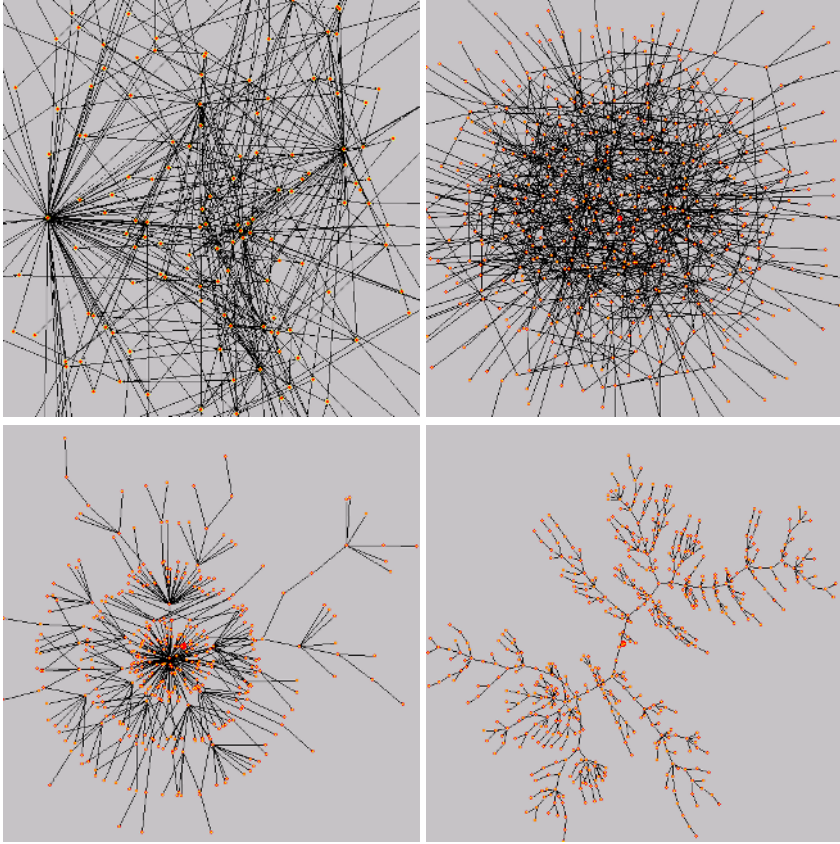


Fig. 1. Top row: Core of the cyclic scale-free Web graph (WG), left, and preferential static network (SN), right. Bottom: Maximum-flow spanning trees of the above graphs, obtained in constant-density traffic $\rho = 1$, with *nnn*-search.

Multiple linking between the same pair of nodes is not allowed. The emergent structure of the static network (SN) where $L = N = 1000$ links is also shown in Fig. 1 (right).

A detailed quantitative analysis of the structure reveals that both incoming and outgoing links behave statistically the same and obey a stretched-exponential distribution. In addition, the clustering in this graph is small compared to the Web graph and link correlations are entirely absent. In Fig. 2 we show the degree distributions for the two graphs.

2.2 Implementation of Constant-Density Traffic

The traffic of information packets on a network is implemented as a *guided random walk* between specified pairs of nodes on the graph [5, 7, 10]—the origin and destination (delivery address) of a packet. The created packets are navigated

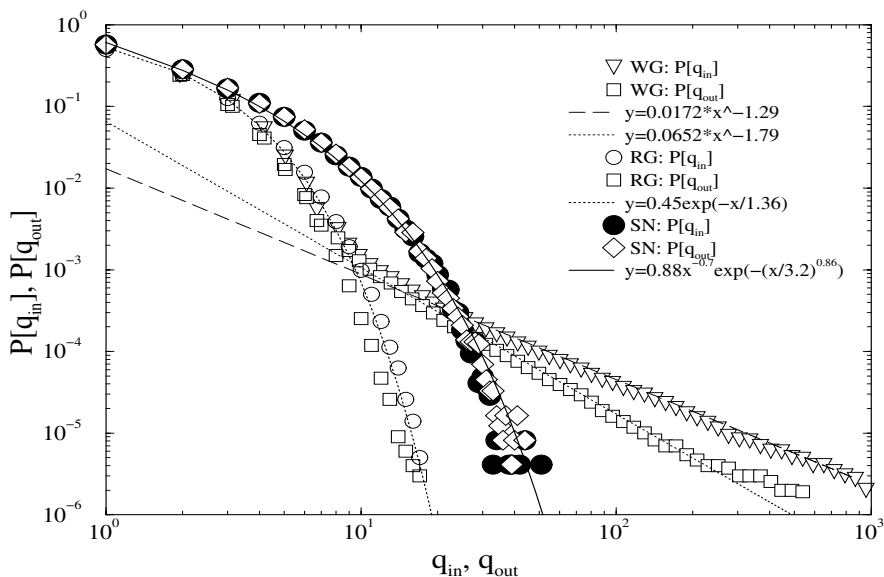


Fig. 2. Degree distribution for incoming and outgoing links in the Web graph (WG) with its power-law tails [9], the static net (SN) exhibiting broad stretched-exponentials, and the random graph (RN), for $N = L = 10^3$ nodes and links

through the graph using the local *nnn*-search rule [5, 10], where two depth levels around each node (sometimes called information horizon 2) are searched for the packet destination address. The rule is supplemented by random diffusion when the search is unsuccessful. The packets are removed when they arrive at their destinations. Here we implement the traffic for a *fixed number of moving packets*. We start with a given number ρ of packets. The arrived and removed packets are replaced in the next time step by creating the same number of new packets at randomly chosen nodes. In the limit $\rho = 1$, this corresponds to the sequential random walk problem. At density $\rho > 1$ packets interact by forming queues at nodes along their paths. We assume finite maximum queue lengths of $H = 1000$, and a LIFO (last-in-first-out) queuing rule. The length of an actual queue at node i we denote by $h_i(t)$. Networks are given by their adjacency matrix. When the graphs are disconnected, as it is the case with both WG and SN, we consider the transport only within their giant clusters.

3 Low-Density Traffic: Limit $\rho = 1$

The limit of *infinitely slow driving* $\rho \rightarrow 1$ is easily realized: as one packet arrives at its destination, another one is created somewhere else in the network. Therefore, packet interaction is absent and the properties of traffic entirely depend on network topology.

By running a large number of packets we record the number of walks along each link (dynamic flow) and through each node (dynamic noise) in the network. Obviously, the inequality of nodes with respect to their local network environment makes the flow and noise fluctuate throughout the network and makes it differ among different network structures. One approach toward a quantitative analysis of these fluctuations is to construct a *maximum-flow spanning tree*, on which each node is connected to the rest of network nodes via its maximum-flow link. Implementing a greedy algorithm, we determine the trees respective to the flow at $\rho = 1$ limit on the two networks. The trees are shown in Fig. 1 (bottom).

The structure of these trees reflects both the underlying network geometry and how that geometry effects transport with given navigation rules—local *nnn*-search. In the case of the WG the tree exhibits a scale-free topology, suggesting a certain degree of compatibility of the traffic and structure. Similarly, for the SN the tree shows some inhomogeneity that resembles the one on the underlying graph.

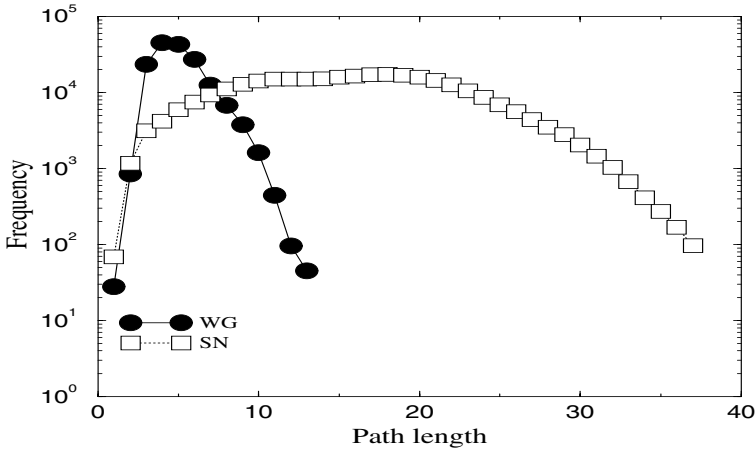


Fig. 3. Distance distributions on the maximum-flow spanning trees for the WG and the SN for $\rho = 1$, and *nnn*-traffic

The max-flow spanning trees represent the union of maximum-flow paths on the underlying network structure. In Fig. 3 we show distributions of lengths of all such paths on the two trees that are shown in Fig. 1. Once again, differences in the graph topologies and thus in their maximum traffic trees manifest themselves in the statistics of the maximum-flow paths. The average distance along such paths on the WG and SN differs by a factor of about 5, the maximum distance by about a factor of 3.

4 Noise Properties for High-Density Traffic

For large packet density $\rho \gg 1$ motion of a packet may be affected by other packets moving through the same node. Queues of packets then occur and a

priority rule sets the order of processing (LIFO as a particular case). Apart from the traffic density, the node surrounding on the network determines the length of the queue at that node. In particular, on inhomogeneous networks, hubs appear to receive more packets compared to other nodes, due to their large connectivity. Since in the algorithm one packet is processed per time step, other packets remain in the queue to be processed later (when no new packet is received). The distribution of queue lengths is therefore reflecting the network structure in a particular way. A snapshot of queue-lengths h , for traffic density $\rho = 100$ in the two network structures WG and SN leads to the distributions shown in Fig. 4.

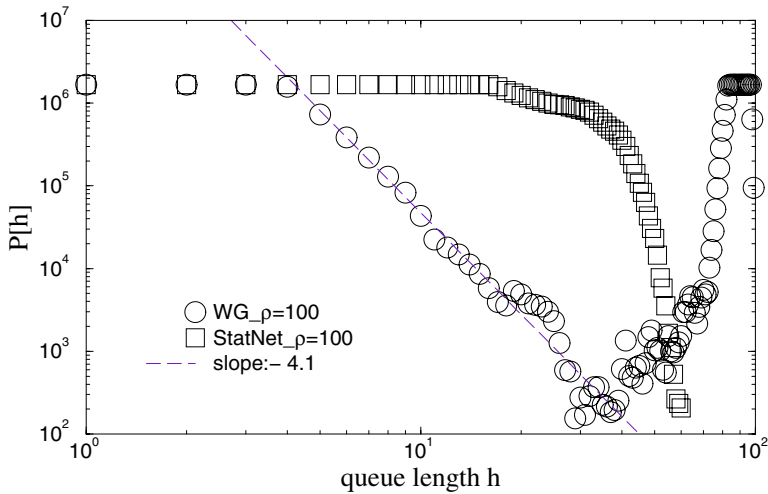


Fig. 4. Distributions of queue-lengths in the WG and the SN for a density $\rho = 100$

In the homogeneous SN most of the nodes are processing a similar numbers of packets, which leads to a flat distribution of queues and a cut-off indicating that queues larger than $h = 40$ are occurring rarely. On the other hand, a large queue of $h = 80 - 90$, packets can be found on the hubs on the inhomogeneous WG with high probability. On the rest of the nodes the queues are distributed with a power-law distribution, apart from very small queues at periphery nodes. The queuing times of packets extend their travel times, thus reducing the overall traffic efficiency [7]. Note, that in the current implementation, with a constant number of moving packets, jamming can not occur as long as the traffic density $\rho < H$, where H is the maximum allowed queue length. However, the travel times of packets can be very long (given by a power-law distribution) [5].

The observed queue lengths are compatible with the the temporal properties of node activity on the two networks, shown in Fig. 5. While queues at important nodes in the inhomogeneous WG are long, the number of nodes that are simultaneously active is small, fluctuating about an average value $n \approx 8$.

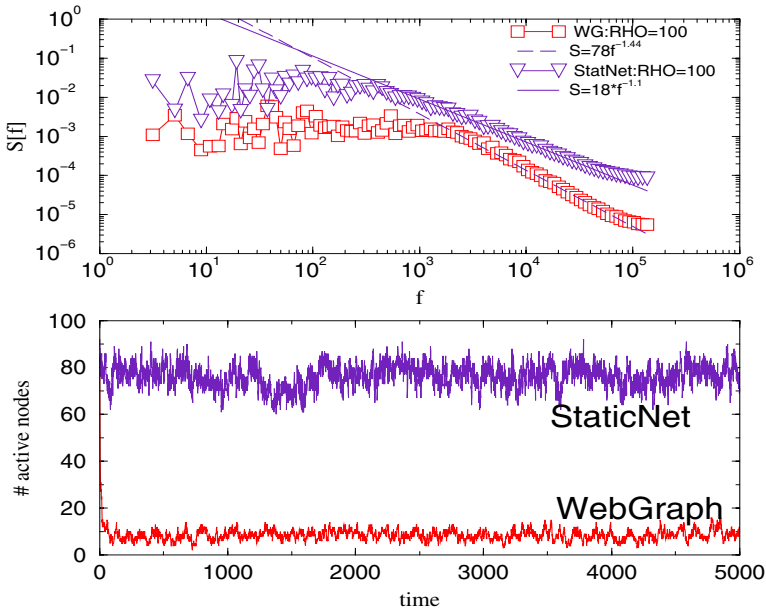


Fig. 5. Temporal fluctuations of the number of active nodes in WG and SN (lower panel) and their power spectra (top panel) for a density of $\rho = 100$ packets and with *nnn*-search

Compared to the more homogeneous SN for the same traffic density, on average $n \approx 80$ nodes are processing a packet simultaneously, leading to short queues at all nodes. Further quantitative analysis of the time series reveals the differences in the packet processing of the two classes on networks. In particular, long-range correlations (anti-persistence) in the number of active nodes develops on both networks. However, in the conditions of constant packet density the fluctuations on the SN appear to be more correlated compared to the WG. The power spectrum exhibiting a power-law behavior $S(f) \sim f^{-\phi}$ for f above some threshold value, are given in Fig. 5 (top panel), with $\phi = 1.1$ for SN and $\phi = 1.4$ for the WG. Therefore, an increased traffic density leads to stronger correlations among node activity on the more homogeneous SN. Differences are also observed in the noise fluctuation properties on these networks, which will be reported elsewhere.

5 Conclusions

We have implemented a constant-density information-packet traffic on two classes of network topologies: correlated, strongly clustered scale-free WG and a more homogeneous, weakly clustered uncorrelated SN. We have shown that for different driving conditions and the *nnn*-search strategy the WG tends to form large queues at its hub nodes, whereas, the more homogeneous SN distributes traffic over a large number of nodes, whose activity appears to be

strongly correlated in time. Therefore despite of generally larger distances as illustrated e.g. with the maximum-flow spanning tree, the homogeneous network performs well at high traffic density.

Acknowledgments. BT thanks for support from the Program P1-0044 of the Ministry of high education, science and technology, Slovenia; ST for the FWF Project P17621, Austria. Partial support by the bilateral project SI-AT/01/04-05 and the COST P10 action is also acknowledged.

References

1. Newman, M. E. J.: The structure and function of complex networks. *SIAM Rev.* **46** (2003) 167-225
2. Boccaletti, S., Latora, V., *et al.*: Complex Networks: Structure and Dynamics, *Physics Reports* **424** (2006) 175-306
3. Guimera, R., Diaz-Guilera, A., Vega-Redondo, F., Cabrales, A., Arenas, A.: Optimal network topologies for local search with congestion. *Phys. Rev. Lett.* **86** (2001) 3196-3199
4. Kaneko, K., *On recursive production and evolvability of cells: Catalytic reaction network approach*, in *Geometric Structures of Phase Space in Multidimensional Chaos: A Special Volume of Advances in Chemical Physics, Part B*, edited by M. Toda *et al.*, **103** (2005) Ch.27, p. 543-598
5. Tadić, B., Thurner, S.: Information Super-Diffusion on Structured Networks. *Physica A* **332** (2004) 566-584; cond-mat/0307670
6. Tadić, B., Rodgers, G.J.: Packet transport on scale-free networks. *Advances in Complex Systems* **5** (2002) 445-456
7. Tadić, B., Thurner, T., Rodgers, G. J.: Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Phys. Rev. E* **69** (2004) 036102
8. Dorogovtsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biology to the Internet and the WWW*, Oxford University Press, 2003. Ch.27, p. 543
9. Tadić, B.: Dynamics of directed graphs: the world-wide Web. *Physica A* **293** (2001) 273-284
10. Tadić, B.: Modeling Traffic of Information Packets on Graphs with Complex Topology. *Proceedings ICCS 2003*, P. Sloot *et al.* Eds., *Lecture Notes in Computer Science*, Springer (Berlin) Part I, **2657** (2003) 136-142
11. Bollobás, B.: *Modern Graph Theory*. Springer (New York) 1998

Local Information Based Algorithms for Packet Transport in Complex Networks

Bernard Kujawski¹, G.J. Rodgers¹, and Bosiljka Tadić²

¹ Department of Mathematical Sciences, Brunel University, Uxbridge,
Middlesex UB8 3PH, UK

{bernard.kujawski, g.j.rodgers}@brunel.ac.uk

² Department for Theoretical Physics, Jožef Stefan Institute, P.O. Box 3000,
SI-1001 Ljubljana, Slovenia
bosiljka.tadic@ijs.si

Abstract. We introduce four algorithms for packet transport in complex networks. These algorithms use deterministic rules which depend, in different ways, on the degree of the node, the number of packets posted down each edge, the mean delivery time of packets sent down each edge to each destination and the time since an edge last transmitted a packet. On scale-free networks all our algorithms are considerably more efficient and can handle a larger load than the random walk algorithm. We consider in detail various attributes of our algorithms, for instance we show that an algorithm that bases its decisions on the mean delivery time jams unless it incorporates information about the degree of the destination node.

1 Introduction

Complex networks can be used to model a wide range of physical and technological systems. One of the most interesting dynamical problems on network is transport, which can give us some insight into the transport of information in technology based communication networks like the internet [1], the World Wide Web [2],[3] or phone call networks [4]. Here we use the term *transport* to mean transport of particles, which are packets in a network. Thus our model falls within the Network Layer of the OSI Reference Model and the algorithms described in section 3 are routing algorithms that belong to the Network Layer of the OSI Reference Model. Of particular interest is the phenomenon of load in a network, as a function of the rate of packet creation R , which has been investigated for models of communication networks [5, 6],[7],[8] and in real networks [9].

Typically the problem of transport is investigated using either a random walk algorithm [6], or the shortest path algorithm used by most internet protocols. The difficulty with these approaches is that random walk algorithm is very inefficient for transport in technology based communication networks and shortest path algorithm requires, for its implementation, information about all connections in network. In this paper we focus on algorithms that use local information about the topology, along with information about the flux of packets between neighbors, the link load and the time taken to deliver packets. We propose four algorithms that use some or all of these properties to deliver packets in a network.

In section 2 we describe the algorithm that we use to perform numerical simulations of our models. In section 3 we discuss the algorithms that packets use to find their destinations and in section 4 we show our results. In section 5 we summarise our results.

2 The Program

A program was written to simulate packet transport on a network that does not depend on the size of the network or its topology. At the beginning of the program an external file with the adjacency matrix of the network is read in. We focus on the internet and consequently we treat nodes in our network as if they were routers. The connections between the routers have the same capacity for all networks. Such a model can not only be used to model internet packet transport but also for a range of transport networks in which the nodes have local routing information.

Each Node

- Generates a new packet with probability $r = R/N$ and with a randomly chosen destination, where R is a fixed rate for the whole network, and N is the number of nodes in network.
- Stores packets in a queue, which has maximum length is $L = 1000$. Packets are despatched from the queue in a first in first out (FIFO) order.
- Sends packets to its neighbours.

Each Node Has Information About

- The address of all its neighbours (they have unique indices j).
- The degree of its neighbours - $k(i)$.
- Flow through all its neighbours, which is measured by
 - The number of packets posted down each edge to neighbour i - the Link Load - $C(i)$.
 - The number of packets sends through neighbour i , which have reached their destination - $N_P(i)$.
 - The sum of the delivery times of all the packets sent through neighbour i that have reached their destination - $T_P(i)$.
 - The time interval since an edge last transmitted a packet to neighbour i and current time step - $\Delta T(i)$.

The index i enumerates each neighbour of node k and each node keeps all the statistics about its neighbours. Quantities $C(i)$, $N_P(i)$, $T_P(i)$ and $\Delta T(i)$ describe node i from the perspective of node k . Each node is described by its neighbours and all properties can be different for all neighbours that describe node i .

The initialization part of the program sets up the network topology, the nodes and all the tables used by them. Inside the main loop a time step is incremented, and within that a loop over all nodes calculates and updates the statistics. The loop over all nodes includes three basics routines, which are run for each node;

generating new packets, checking its queue for packets with its address and sending packets to its neighbours. Each node generates a packet with a randomly chosen destination with probability R/N . The node checks its own queue for packets addressed to itself. When it finds one of these it deletes it from the queue and updates the statistics $N_P(i)$ and $T_P(i)$ for all the nodes on the packet's path. Each packet keeps track of its own path. The node sends packets to its neighbours by taking the first packet in its queue and checking the packet destination address. If the packet is addressed to one of its neighbour, the node will send it to the neighbour. If it is not, the node will use the *algorithm* to find where to send the packet. During this posting step the $C(i)$ property is updated. When node k sends packets to node i , the number of sent packets $C(i)$ increases. After this loop over all the nodes is completed the quantities $\Delta T(i)$ and the mean delivery time of packets sent down each edge $N_P(i)/T_P(i)$ are updated for all nodes.

3 Algorithms

The most important element in transport is the rule that determines the direction in which a packet is sent. A transport network without a rule is a random walk network. We call this rule the *algorithm*. It describes how nodes deal with packets and should help packets to get to their destination. Not all algorithms help packets to reach destinations, poor algorithms can easily be worse than the random walk algorithm. All algorithms considered in this paper work with deterministic rules.

The *shortest time* (ST) algorithm is our basic algorithm that uses information about the mean delivery time $T_P(i)/N_P(i)$ and the time interval between the last packet that came to node i and actual time step. The ST algorithm finds the minimum value

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} \right]_{i=1 \dots n} \quad (1)$$

in order to determine which node to send the packet to. The idea of this algorithm is to try and find the minimum travel time for each packet between source and destination. At the start of the simulation S is equal to 0 for all neighbours. Because the update of $T_P(i)/N_P(i)$ only occurs when a packet arrives at its destination, it can take a number of time steps before $T_P(i)/N_P(i)$ becomes non-zero. The inclusion of the reciprocal of $\Delta T(i)$ in S ensures that the algorithm does not get into a state where it never sends a packet down certain links which have a large mean delivery time. This state is particularly likely to occur at the start of the simulation. The inclusion of the reciprocal of $\Delta T(i)$ in S also prevents overcrowding when a node finds a node which is clearly better than all its other neighbours. Hence, because of the inclusion of $\Delta T(i)$ more nodes take part in the transport and in this way the large node do not become overcrowded. Because the algorithm with $T_P(i)/N_P(i)$ is looking for minimum delivery time we call it the *shortest time* (ST) algorithm. To start this algorithm, and the STD algorithm, which we will introduce shortly, we use the random walk algorithm. We only use

the deterministic algorithms at a node when all the values of S of its neighbours are greater than 0. Without this initial random walk procedure both the ST and the STD algorithms would jam almost immediately. The *shortest time and degree* (STD) algorithm is a modification of the ST algorithm. It uses information about the local topology, the degree. This helps packets avoid the nodes with the largest degree, which are mostly overcrowded. The idea of incorporating information about the degree of nodes in the transport algorithm was discussed in [10] and [11]. In these papers models were introduced in which nodes were selected at a rate proportional to a power of their degree. It was found that the most efficient algorithm was one in which the probability of selecting a node of degree k was proportional to $1/k$ [10] and [11]. The STD algorithm is defined by

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} k(i) \right]_{i=1\dots n} \quad (2)$$

where $k(i)$ is a degree of node i and $k(i) > 1$. This last assumption allows the algorithm to avoid dead-end nodes. A node with degree $k = 1$ can only receive a packet that is addressed to itself. The STD algorithm uses both temporal properties and also information about the local connectivity. For transport in a scale-free network the most important nodes are those with the largest degree. But because their neighbours send these nodes a large number of packets the queues at these nodes can become overcrowded. Information about the degree helps the algorithm to avoid these nodes, but it does not mean that they are not used.

The *connections and degree* (CD) algorithm and the *connections, degree and shortest time* (CDT) algorithm use information about the link load $C(i)$. Because of this the random walk starting procedure used in the ST and STD algorithms is not required for the CD and CDT algorithms. The CD algorithm uses only information about the link load and the degree. The CD algorithm is defined by

$$S_k = \min[C(i)k(i)]_{i=1\dots n} \quad (3)$$

where $C(i)$ is a number of packets that node k sends to node i .

For this algorithm S equals 0 at the start, but $C(i)$ is updated almost immediately. When node k sends a packet then it automatically increases the value of $C(i)$. There is no need to wait for information from the destination about the delivery time like in the ST and STD algorithms. In this way CD algorithm improves very quickly and the random walk is not needed. The link load, $C(i)$, quantity helps the algorithm to deliver packets and ensures that almost all nodes take part in the transport. The degree quantity helps to prevent the largest nodes from becoming overcrowded. In this algorithm there is no property that can be optimised, unlike in the ST and STD algorithms where the delivery time is optimised.

The CDT algorithm is intermediate between the CD and the ST algorithms. It optimises the delivery time and does not need the random walk starting procedure because it includes a dependance on the link load, $C(i)$. The dependance on degree prevents large nodes becoming overcrowded. For the CDT algorithm,

the starting procedure is the same as for the CD algorithm except that we set the ratio

$$\mathcal{R} \equiv \frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} ; \quad (4)$$

equals to 1 at the start to avoid 0 value. This means that we do not need to start off with a random walk algorithm as in the ST and STD algorithms. The CDT algorithm is defined by

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} C(i) k(i) \right]_{i=1 \dots n} \quad \text{with } k(i) > 1. \quad (5)$$

We use the learning property to describe behavior of an algorithm in the beginning. By learning we mean the proportion of links whose value of S has changed since $t = 0$. The CD and CDT algorithms learn the most quickly. After 5000 time steps they tried 95% of links. This is because the link load, $C(i)$, changes when a packet is sent down it whereas $T_P(i)/N_P(i)$, used by the ST and STD algorithms, only changes when a packet sent down it gets to its destination. That is way the ST and STD algorithms need the random walk starting procedure. With this procedure after 5000 time steps 35% of links were tried. For the ST algorithm without the random walk starting procedure it was 5%. The speed of learning is important because when a network learns slowly, the network only uses a small proportion of its links for transport over a long period of time, which means that the network is easily jammed when a region of the network becomes overcrowded.

4 Results

We consider transport on the scale-free network [12] with $N = 1000$ nodes and $m = 2$ outgoing links per node. This is an example of uncorrelated network with a power-law distribution of incoming links $P(k) \sim k^{-3}$ for large k . When $m = 2$ the network includes loops and has relatively small number of connections. Our research show that this network jams for lower values of the posting rate than networks with $m = 1$ or $m = 3$ and higher. In this work we use a posting rate of $R = 0.1$. This means that each node creates a packet with probability R/N . The number of time steps for all our simulations is 500,000. We present results for the STD, CD and CDT algorithms. We do not consider the ST algorithm any further because it isn't stable and always jams.

In figure 1a we show the load in the network, the number of packets that are still in the network. All three algorithms are stable (exhibiting stationary flow). We compared the level of load by finding the mean value of the number of packets in the network. The best algorithm with smallest mean value is the STD algorithm. For the CD and CDT the values are almost the same.

The number of packets in network can be treated as a noise in the network. Measuring the power spectrum of the noise shows that there are correlations in the number of packets on the network. For all three algorithms the power

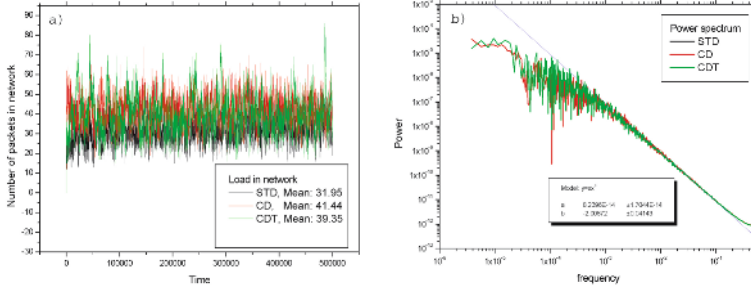


Fig. 1. The load properties. a) Load in the network against time for the STD, CD and CDT algorithms. b) The power spectrum of the load time series.

spectrum (Fig. 1b) has the slope $1/f^{-\phi}$ with $\phi = -2$, indicating short-range correlations.

Further we measured the time interval $\Delta T(i)$, the time that a node waits for next packet. The results for the distribution of $\Delta T(i)$ are shown in figure 2 for all three algorithms. This quantity is important for the improved navigation in the STD and CDT algorithms. We found that without the $\Delta T(i)$ in Eqs (2, 5) the networks easily jam.

For the STD algorithm the distribution of $\Delta T(i)$ has a tail and on a double logarithmic scale has a slope $b = -3/2$. The cut-off comes from the finite time of the simulation. The first part of the distribution for all algorithms is flat. For the CDT algorithm the function falls faster than for the STD. This is connected with the inclusion of the link load in the CDT algorithm, which means that more links are used and long time intervals of $\Delta T(i)$ do not occur as frequently as in the STD algorithm. The CD algorithm does not make use of $\Delta T(i)$ but we measured it for comparison.

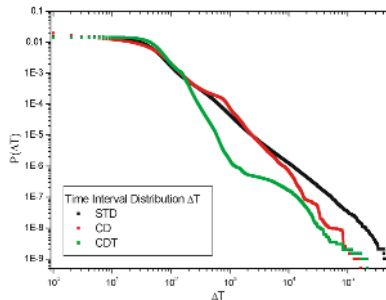


Fig. 2. Distribution of time interval $\Delta T(i)$ for three navigation algorithms

The distribution of packet delivery time T are given in figure 3a. At long times these distributions appear to be similar for all three algorithms. However the number of packets delivered in a short time differs between the algorithms.

In particular, for the STD algorithm the probability that a packet is delivered quickly is much higher compared to the CD and CDT algorithms. This is reasonable, because STD algorithm finds the paths with the shortest delivery time. Whereas, the CD and CDT algorithms are distributing the transport across the network making use of the link load $C(i)$ condition. The distribution $P(T)$ for the CDT algorithm interpolated between the STD and CD algorithms reflects its dependence on both the link load $C(i)$ and the shortest time statistics.

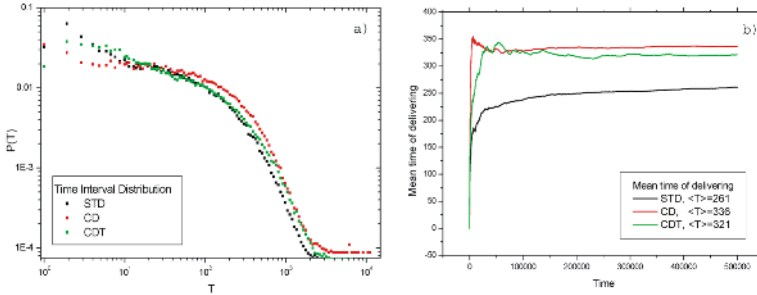


Fig. 3. The time delivery quantities. The distribution of delivery times of packets (a) and the average delivery time (b) for the STD, CD and CDT algorithms.

The time series for the overall mean delivery time of packets (Fig. 3b) show further differences between the navigation algorithms. The algorithms involving the delivery time statistics for $T_P(i)/N_P(i)$ reach a stable (optimal) level of the average delivery time and maintain it almost constant. On the other hand, the mean delivery time in the CD algorithm increases slowly, suggesting that in this case no optimisation is taking part since none of the involved quantities is requested to be optimized.

The observed result for the STD and CDT algorithms arise through two effects: First, inserting the $\Delta T(i)$ condition in the search rule, nodes intend to send packets through rarely used links. Often these are not the best choices for the transport, however, this mechanisms prevent the network from jamming. Second, including the degree rule in the navigation, nodes preferably send packets to neighbours with a small degree, which potentially makes the delivery times longer on the scale-free network.

5 Conclusions

We have introduced several navigation algorithms that are capable to improve the packet transport by optimizing the average delivery times of packets and preventing the network from jamming. We have demonstrated comparatively how these algorithms work in traffic on an uncorrelated scale-free network, which is known to be very prone for jamming [5]. These are the algorithms STD and CDT, which are based on the updated statistics of the travel times of packets (global information) and (local) information on the degree of their nearest

neighbour nodes. For comparison, we have also shown how the traffic behavior changes when parts of these informations are missing, as in the CD and ST algorithms, respectively. In the absence of local rule, the ST navigation leads to traffic jamming relatively quickly, although it is better compared to the random diffusion for the same posting rate. Whereas, in the case of navigation with the CD algorithm (no global information), the traffic seems to be stationary and the results comparable with the other two methods. When the shortest time property is used it needs to be balanced by the degree rule on scale-free networks. The existence of hubs causes traffic congestion for the shortest time algorithm, similarly to the traffic along the topologically shortest paths.

On the technical level, implementation of the STD and CDT algorithms the problem is in finding the accurate value for the edge dependent properties. A node needs a lot of traffic through a given link in order to find its proper time statistics. Because the mean delivery time is very long, it takes a lot of time to set up the edges dependent properties for all nodes. In particular, the algorithms that depend on the time $\Delta T(i)$ and the degree $k(i)$ do not jam but the costs is in learning phase and therefore increased mean delivery time. Using the local property $\Delta T(i)$ in the navigation rule helps in preventing the jamming but at the same time deteriorates the feedback effects from the travel time statistics. The degree property helps the algorithm to avoid nodes with large degree, but it also results in long delivery times. Our results show that in scale free networks we cannot avoid using nodes with large degree.

In the future work the methods developed here can be used for determining the optimal transport path for packets. Some of the related subjects are applications of these algorithms to different or more realistic network geometries, varying posting nodes, and increased traffic “bandwidths” at large node.

References

1. Faloutsos, M., Faloutsos, P. and Faloutsos, C.: *Comp. Comm. Rev.* **29** (1999) 251
2. Albert, R., Jeong, H. and Barabasi, A.-L.: *Nature* **401** (1999) 130
3. Huberman, B. and Adamic, L.: *Nature* **401** (1999) 131
4. Adamic, L. A. , Lukose, R. M., Puniyani, A. R. and Huberman, B. A.: *Phys. Rev. E* **64** (2001) 046135
5. Tadić, B. and Thurner, S.: *Physica A* **332** (2004) 566
6. Tadić, B. Thurner, S. and Rodgers, G.J.: *Phys. Rev. E* **69** (2004) 036102
7. Arenas, A., Diaz-Guilera, A. and Guimera, R.: *Phys. Rev. Lett.* **86**(2001) 3196
8. Sole, R. and Valverde, S.: *Physica A* **289** (2001) 595
9. Jacobson, V.: in *Proceedings of SIGCOMM '88* (ACM, Standford, CA, 1988)
10. Yan, G., Zhuo, T., Hu, B., Fu, Z.-Q. and Wang, B.-H.: *cond-mat/0505366* (2005)
11. Yin, C.-Y., Wang, B.-H., Wang, W.-X., Zhou, T. and Yang, H.-J.: *cond-mat/0506204* (2005)
12. Albert, R. and Barabasi, A.-L.: *Rev. Mod. Phys.* **74** (2002) 47

Empirical Analysis of the Spatial Genetic Algorithm on Small-World Networks*

Yong Min¹, Xiaogang Jin^{1,2,**}, Xianchuang Su³, and Bo Peng¹

¹ AI Institute, College of Computer Science, Zhejiang university,
Hangzhou 310027, China

² Ningbo Institute of Technology, Zhejiang university, Ningbo 315100, China
xiaogangj@cise.zju.edu.cn

³ College of Software Engineering, Zhejiang university, Hangzhou 310027, China

Abstract. Genetic algorithm (GA) has been widely used in optimizing and solving various problems since first proposed, and its characters also have been deeply studied. In this paper, we investigate the benefits of genetic algorithm whose population is distributed on small-world networks. In particular, we pay our attention to the complexity of how small-world affects the behavior of spatial GA. Our work shows that, on a complex problem, the behavior of spatial GA on the small-world networks is influenced by at least two different factors: local selection and asymmetric topology. It is more complex than previous results from simple lattice models. Our results could provide lots of potential methods to improve the performance of spatial GA and give some guidance for designing of parallel genetic algorithm. We also present many future problems on the influence of small-world to spatial GA.

1 Background

1.1 Spatial GA and Local Selection Effect

Influence of spatial structure on evolution has been researched from various fields: biology[1][8][10], mathematics[6] and computer science[9][11]. These researches have proved that the spatial structure would deeply affect evolution. Spatially structured population has been also proposed in genetic algorithm as a tool for improving the searching properties of GA. Spatial GA is a kind of genetic algorithm whose population has certain spatial structure. The previous studies on spatial structure of population were primarily related to the parallelism genetic algorithm (PGA)[3][5]. Here, we should mention about fine grained parallel genetic algorithm: the population is distributed on a network, therefore, one individual and his directly connected neighbors form a deme. Any individual could only interact with their neighbors, when we perform operators of GA, such as selection or mutation. The authors noted that the performance of the algorithm

* Supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. Y105697 and Ningbo Natural Science Foundation (2005A610004).

** Corresponding author.

degraded as the size of the deme increased. This effect, called local selection, has been researched further by Jayshree Sarma and Kenneth De Jong[7]. Their results show that the critical parameter of spatial GA is the ratio of the radius of the neighborhood to the radius of the underlying networks. Moreover, the coefficient of the growth rate of the best individual in the population is shown to be an inverse exponential function of this ratio.

1.2 Small-World

The networks that occur in lots of social, biological, and man-made systems are usually neither completely regular (such as lattices), nor completely random. These networks usually possess two characters: one is average vertex-to-vertex distances increase only logarithmically with the total number of vertices, another is so-called “clustering” that two neighbors of a vertex will often also be the neighbors of another. The networks that have above two characters are named “Small-World”. In order to study the spatial GA on the small-world networks, we need a artificial model of small-world networks. Watts and Strogatz[2] have proposed an famous model for the small-world (WS model), which perhaps fits better than real small-world networks. Fig.1.1 and Fig.1.2 schematically describes the process for building small-world of WS model. The details of WS mode can be found at [2].

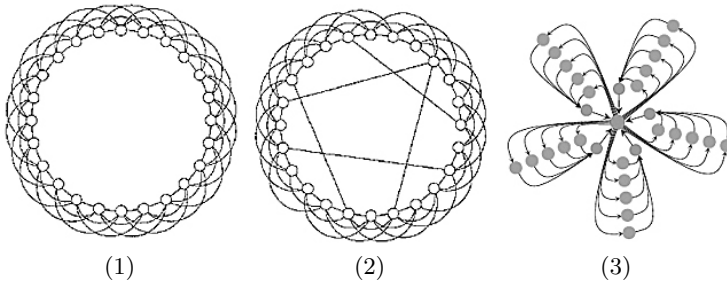


Fig. 1. The Watts-Strogatz Model. AND Super-Star Structure.

1) The regular lattice with periodic boundary conditions, so that the system becomes a ring. Each vertex connected to its z nearest neighbors, where in this case $z=6$ (radius of the neighborhood r_n : $2r_n = z$). 2) The Watts-Strogatz model is created by rewiring a small fraction of the links (in this case five of them) to new sites chosen at random. 3) The super-star structure.

1.3 Asymmetric Topological Structure

More recent discoveries of spatial influence of evolution, from Erez Lieberman *et al.*[4], suggest that the amplification of selection occurs at a large amount of vertices feeds into single vertices which then feed into possibly further single vertices until they feed into the central hub and finally the hub feeds back into the these vertices. This feedback loop acts as a filter that reduces stochastic effect of

random drift. Fig.1.3 illustrates a such structure[4]. Generally, in a network, we define that the sum of weight of any vertex's leaving links (W_{out}) is 1. Therefore, for a vertex, if sum of weight of entering links (W_{in}) is equal to 1, this vertex neither suppress nor amplify selection; if $W_{in} < 1$, this vertex amplify selection; if $W_{in} > 1$, this vertex suppress selection. In sum, the asymmetric topology could affect the behaviors of evolution.

In this paper, it is the first time that explores the effects of asymmetric topology for genetic algorithm. We also want to illustrate that the behavior of spatial GA on small-world is complicate, which is affected by at least two factors: local selection and asymmetric topology.

2 Experimental Setup

2.1 Algorithm

Our experiments base on Matlab's Genetic Algorithm and Direct Search (GADS) toolbox, and we modify its algorithm to fit for our requirements. The more details of Matlab's GADS toolbox can be found in Matlab's documents and website[12].

In our GA (we call it NGA: genetic algorithm on networks) which is similar with grained parallel GA, but it run on a single machine, the population is distributed on small-world networks generated by WS model. NGA has three main steps which are differ from standard genetic algorithm(SGA: Standard algorithm in Matlab): First step is preserving several elites to next generation according to the ranking of individuals' fitness. One individual just preserve one time. The preserved individuals will be stored in the same position at network as their current position. Second step is generating mutation offspring whose number is m . In the step, we first choose m unoccupied vertices from underlying network, which are labelled by $v_i, (i = 1, 2, \dots, m)$. For each v_i , we should choose a parent from the individuals in v_i and its neighbors, and the mutation of the parent will occupy v_i in next generation. Third step is generating crossover offspring. Generating method likes method described in second step, but it needs two parents for one offspring and the offspring is generated by crossover operator. The main difference between NGA and SGA is whether the sequence or position of individuals can be ignored. SGA's population is a simple set of individuals, but structure of population of NGA is corresponding to the underlying network structure.

2.2 Definition of Success and the Diversity of Population

At the end of algorithm running, the best individual's fitness is f_e , and the target fitness is f_t . In our experiments, the value of fitness is just the value of target functions. We say running of algorithm is success, if $|f_e - f_t| < 0.01$.

In our NGA, the coding space is $\{0, 1\}^L$, size of population is n . The population is $P = \{a_1, a_2, \dots, a_n\}$, where $a_j = (a_{1j}, a_{2j}, \dots, a_{Lj}), j = 1, 2, \dots, n$, so, the diversity of population is defined as:

$$D(P) = 1 - \frac{1}{L \cdot n} \sum_{l=1}^L (\max\{\sum_{j=1}^n (a_{lj}), \sum_{j=1}^n (1 - a_{lj})\} - \min\{\sum_{j=1}^n (a_{lj}), \sum_{j=1}^n (1 - a_{lj})\});$$

obviously, $0 \leq D(P) \leq 1$. When $D(P) = 0$, the diversity of population is minimal, and it is maximal, when $D(P) = 1$.

2.3 Experiment 1

In order to ensure that the results of our experiments are correct, we need to prove NGA is “equivalent to” SGA, when NGA plays on a fully connected network. It is imply that the improving or worsening comes from small-world effect but from program. Here, we use Rastrigin’s function, which is commonly used for testing genetic algorithm, to test both NGA on fully connected networks and SGA. The definition of two-dimensional Rastrigin’s function is: $R_1(x, y) = 10 \cdot n + [x^2 - 10 \cdot \cos(2\pi x)] + [y^2 - 10 \cdot \cos(2\pi y)]$, and the target task is: $\min(R_1)$, where $x, y \in [-5.12, 5.12]$.

The NGA has been defined above, and SGA come from the Matlab toolbox. Main parameters for Matlab’s GA algorithm are presented in below table:

Parameter Name	Value
Population Type	bitstring (length is 18)
Elite Count	10 percents of population
Crossover Probability	80%
Maximal Generation	100
Selection Method	roulette
Crossover Method	single point

what is meaning of these parameters and how they affect the behavior of algorithm can be found in Matlab documents[12]. Other default parameters also can be found in the documents. Every algorithm will repeat 100 times independently, and we will compare NGA with SGA at these parameters: success rate, diversity of population and convergent speed. From those, we want to prove that NGA is “equivalent to” SGA.

2.4 Experiment 2

In this experiment, we want to explore how small-world affects the performance and behavior of spatial GA. Therefore, we will execute NGA on a series of small-world networks of WS model. These small-world networks base on one-dimensional lattice that have 256 vertices and $r_n = 2$. The rewiring probability p_i , ($1 \leq i \leq 29$) of those networks is:

$$p_i = \begin{cases} (i-1)/256 & ; \quad 1 \leq i \leq 21 \\ (i-21)/10 & ; \quad 22 \leq i \leq 29 \end{cases}$$

Therefore, there are 29 different networks: G_i . G_i ’s rewiring probability is p_i , where $1 \leq i \leq 29$.

We choose Rosenbrock’s function: $R_2(x, y) = 100 \cdot (y - x^2)^2 + (x - 1)^2$ for fitness function, and target task is: $\min(R_2)$, where $x, y \in [-5.12, 5.12]$. (SGA

performs badly in Rosenbrock’s function. Therefore, it fits for describing the improvement and the properties of NGA on small-world.) The parameters of algorithm in experiment 1 and 2 are identical except the maximal generation is 200 in this experiment and every algorithm will repeat 120 times independently.

In regular lattices, the degree of any vertex is identical. It is 4 in this experiment. By rewiring, many vertices’ degree would be bigger than 4, and some others would be smaller than 4. Here, vertices are labeled by $v_i, 1 \leq i \leq 256$, and the degree of vertices is labeled by $degree(v_i)$. we define asymmetric factor U as:

$$U = \sum degree(v_i), \text{ where } degree(v_i) > 4, 1 \leq i \leq 256$$

Although this is a crude definition for asymmetry, it would be effective in our experiment.

Comparing with NGA on small-world, we also use same experimental conditions to test SGA and NGA on regular lattices (the size of one-dimensional lattices is also 256, and $1 \leq r_n \leq 128$).

3 Results and Analysis

3.1 Experiment 1

The results of experiment 1 are presented in Fig.2.1 and Fig.2.2. Two curves in the figure almost coincide, and success rate of NGA and SGA is 93% and 91% respectively. Therefore, we can imply NGA is “equivalent to” SGA, when it base on the fully connected networks. According to the results, we can ignore the influence of program in next experiments.

3.2 The Benefits of Small-World

The “success rate” is the percentage of successful runs. The success rates of different algorithms are presented in below table:

Algorithm	Success Rate
NGA on small-world (Best)	0.6750
NGA on small-world (Worst)	0.4083
NGA on regular lattice ($radius = 2$)	0.5750
NGA on regular lattice (Best, $radius = 4$)	0.6917
SGA	0.3858

From the table, we can find: 1) the worst performance of NGA on small-world is even better than the performance of SGA; 2) the best performance of NGA on small-world is much better than the performance of NGA on regular lattice on which small-world model base; 3) the best performances of NGA on regular lattice and on small-world are approximate sameness, but, comparing with regular lattice, average connected degrees reduce by half in small-world model, For

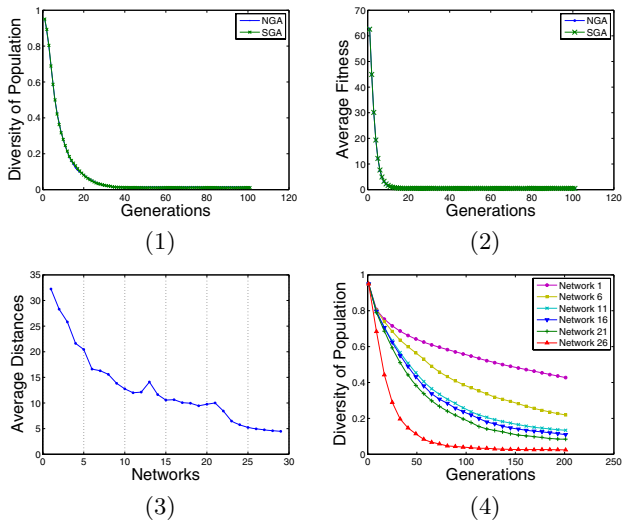


Fig. 2. Results of Experiment

(1) is graph of the average diversity of population at each generation. (2) is graph of average fitness of population at each generation (mirror the convergent speed of GA). (3) is graph of average distances for all 29 networks. (4) is graph of diversity of population at each generation for $G_1, G_6, G_{11}, G_{16}, G_{21}, G_{26}$.

parallel GA, it means that the cost of communication of network will decline dramatically.

In sum, the small-world plays an extraordinary role in promoting the performance of genetic algorithm.

3.3 Relationships Between Average Vertex-to-Vertex Distances and the Diversity of Population

The relationships between average vertex-to-vertex distances (labeled by L) and diversity is described by Fig.2.3 and Fig.2.4. There are two obvious factors: First, the changing rate of diversity increase with the decreasing of L ; Second, L of G_{11}, G_{16} and G_{21} is similar, at the same time, the diversity curves of G_{11}, G_{16} and G_{21} is also close to each other. Therefore, we can deduce that L would determine the changing rate of diversity of population in spatial GA. It is reasonable. Shorter L would promote individuals' diffusion. It let a few advantage segments of "gene" occupy more and more individuals in the population, and disadvantage segments of "gene" are replaced rapidly and disappear ultimately.

3.4 Different Results from Previous Researches

Jayshree Sarma and Kenneth De Jong's results[7] show that the critical parameter, which deeply affects the behavior of spatial GA, is the ratio (labeled by R) of

radius of neighborhood to the radius of the underlying networks. However, from Fig.3, we find two opposite phenomenons: First, trends of two curves (Fig.3.2) aren't consistent totally; Second, the turning-points of two curves cannot match to each other. Although the improving from local selection is clear from above results, the two factors would support our opinion that R cannot entirely determines the behavior of spatial GA on small-world, on the contrary, the crucial role of R in evolution on regular lattices is obvious.

3.5 New Discovery: The Influence of Asymmetric Topology

From Fig.3.1, we can find an interesting phenomenon: two curves' turning-points are well matched to each other (these points include: 8, 9, 10, 11, 16, 17, 19, 20, 21), and there are just a few unmatched points, such as 15 and 22. All of those cannot be explained by contingency. Therefore, we could say asymmetric topology of underlying networks could also affects the behavior of spatial GA on small-world. I concede that two curves are unmatched at tail, however, I think the most possible reason is our asymmetric factor U cannot correctly reflect the degree of asymmetry of small-world, when there are too many rewired links. In sum, the results of our experiments suggest that the influence of asymmetric topology cannot be ignored, when you design a spatial GA. Additionally, the performance of spatial GA might be further improved by well designed asymmetric topology.

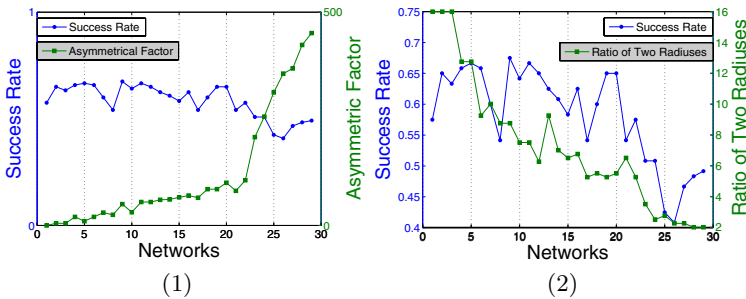


Fig. 3. Results of Experiment 2

(1) is the graph of asymmetric factor for all 29 networks; (2) is the graph of ratio of the average connected radius of the vertex to the radius of the underlying networks for all 29 networks.

4 Conclusion and Future Works

In our experiments, spatial GA on small-world exhibit its outstanding benefits, comparing with SGA and spatial GA on regular networks. In common sense, these benefits seem to be due to local selection effect. Furthermore, we have found that another factor, asymmetric topology of networks, also play a subtle role in

the spatial GA, and we have given a few evidences to support this discovery. The discovery could provide us a potentially valuable tool to improve the performance of spatial GA, and also helps us to understand the behaviors of evolution on small-world.

Admittedly, our work is just a beginning of exploring spatial GA on small-world networks. In the future, more works need to be done to analyze the properties of the behavior of spatial GA on small-world, such as more details about asymmetric topology, relationships between local selection and asymmetric topology on small-world, and how to develop more efficient genetic algorithm with asymmetric topology.

References

1. C. Hauert, M. Doebeli: Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*, **428** (2004) 643-646
2. D. J. Watts, S. H. Strogatz: Collective dynamics of 'small-world' networks. *Nature*, **393** (1998) 440-442
3. E. Cantu-Paz: Efficient and Accurate Parallel Genetic Algorithm. Kluwer Academic Press (2000)
4. E. Lieberman, C. Hauert, M. A. Nowak: Evolutionary dynamics on graphs. *Nature*, **433** (2005) 312-312
5. E. Alba, M. Tomassini: Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, **6**(5) (2002) 443-462
6. H. Ebel, S. Bornholdt: Coevolutionary games on networks. *Phys. Rev. E*. **66** (2002) 56-118.
7. J. Sarma, K. De Jong: An analysis of the effects of neighborhood size and shape on local selection algorithms, *Parallel Problem Solving from Nature IV*, Berlin, (1996) 236-244
8. M. Nakamaru, H. Matsuda, Y. Iwasa: The evolution of cooperation in a lattice-structured population. *Journal of theoretical Biology*, **184** (1997) 65-81
9. M. Giacobini, M. Tomassini, A. Tettamanzi: Takeover time curves in random and small-world structured populations. *GECCO'05*, June 25-29, 2005, Washington, D.C., USA
10. M. A. Nowak, R. M. May: The spatial dilemmas of evolution. *Int. J. Bifurcation Chaos*, **3** (1993) 35-78
11. N. Williams, M. Mitchell: Investigating the success of spatial coevolution. *GECCO'05*, June 25-29, 2005, Washington, D.C., USA
12. The Matlab's genetic algorithm and direct search toolbox documents: <http://www.mathworks.com/access/helpdesk/help/toolbox/gads/>

An Evolution Process Model for the Internet Topology*

Sangjoon Park¹, Insook Cho², and Byunggi Kim²

¹Information & Media Technology Institute, Soongsil University

²School of Computing, Soongsil University
{lub, bgkim}@archi.ssu.ac.kr

Abstract. Instead of actual experiments to network protocols, network simulators are useful to analyze these network protocols for lower analysis cost. The Internet topology is dynamically evolving and growing, and then shows changing characteristics based on time flow. Studies of Internet topology have been motivated by the demands for analysis and simulation to the modeling of real networks. Hence, to develop the Internet simulator, proper characteristics to Internet topology should be studied. In this paper, we propose topology models to the Internet topology showing node addition and deletion.

1 Introduction

Real experiments to large scale network are ideal to correctly analyze performance evaluations to researched network protocols. However, it is very difficult since researched protocols cannot be directly tested in real networks, and the evaluation cost based on real experiments is very high. Hence, the modeling and the simulation to the large scale network such like Internet are useful to assess researched protocol performances. Advantages of the modeling and the simulation for such network structure are as follows:

- Evaluation cost:* Generally, the experimental evaluation to the actual network is very difficult and expensive. However, by using a simulator to the source network system, it can spend much lower evaluation cost compared with actual experiments.
- Flexibility:* The experimental implementation has the limitation of application flexibility caused by environment problems when an additional scheme is newly adapted to the source network systems. In simulation environments, the addition of a new scheme is much easier than it on the experimental environments.
- Applicability:* Network simulators developed in the modeling can be used to analyze variety network mechanisms.

The Internet was developed for military and research-oriented prototypes, but it has grown into very large scale network that connects many computers around the world without any centralized control or administration. Hence, to implement simulations related to network protocols on the Internet, it is important to study the properties of and policies of the Internet (e.g., network scale, connection types between different

* This work was supported by the Soongsil University research fund.

networks and network evolution model). Therefore, most researches to the Internet focus topology models that try to represent the characteristics of the real Internet topology accurately [1] [2] [7]. In this paper, we propose an Internet topology model that has the functionalities of adding and deleting nodes and processing the isolated nodes to accurately apply the changes in the real Internet. We consider a node change model for showing characteristics of Internet evolution.

2 Internet Topology Property

2.1 Evolution Model

There are power-laws among the several properties that characterize the Internet topologies [5], [6], [9]. At present, the well-known key power-laws are the following: First, the power-law relationship between the degree of a node (d_v) and the rank of a node (r_v) is presented by

$$d_v \propto r_v^R \tag{1}$$

where v is a node in the Internet, and R is the power of a constant (rank exponent).

Secondly, a power-law appears between an degree (d) and the frequency (f_d) as follows:

$$f_d \propto d^O \tag{2}$$

where O is the degree exponent..

Thirdly, a power-law is the relationship between a calculated eigen value (λ_i) and a rank (i) as follows:

$$\lambda_i \propto i^\varepsilon \tag{3}$$

where ε is the eigen exponent.

The rank is arranged in ascending order by the eigen value using a close matrix. If all nodes are connected, it is represented by 1, and if not connected, it is expressed by 0.

2.2 Topology Model

Waxman model is the distance-based connection that links of nodes are added to a network considering the distance between nodes arranged in random positions on 2D grid [10]. The rate of generating link between the two nodes u and v is different depending on the distance of the node from the node, and it is determined using the connection rate as follows. Waxman model is usually used to create a random network.

$$P(u, v) = \alpha e^{-\frac{d}{\beta L}} \tag{5}$$

where d is euclidean distance from node u to v , L is the maximum distance between two nodes in a graph, and $0 \leq \alpha, \beta \leq 1$.

Barabasi-Albert model is the degree-based connection method that a characteristic of the Internet evolution shows a power-law distribution by providing incremental growth and preferential connectivity to the degree frequency [3][4].

$$P(j) = \frac{d_j}{\sum_{k \in V} d_k} \tag{6}$$

where d_j is the degree of the node (j) and V is the collection of all nodes connected in the network. $\sum_{k \in V} d_k$ is the degree sum of all existing nodes.

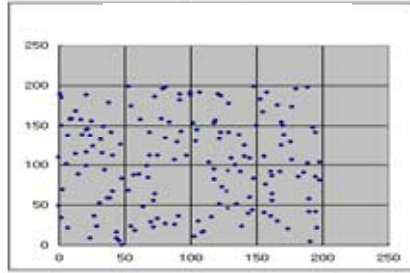


Fig. 1. Random topology (Arrangement of nodes without links)

3 Internet Topology Model

In this paper, we design an Internet topology model showing changeable network configuration. In the time flow, network nodes can be deleted or added to the growing Internet topology.

3.1 Method to Make a Topology in the Beginning

Firstly, to present the initial Internet topology, we use the node information (1999 year-network information) to the Korea Internet network. The node generation pattern is that the initial node is generated and then will be arranged in the network. Furthermore, after creating a node in small-sized topology, the arrangement to the node uses a random method. A link connection between two nodes uses the node connection rate [equation (5)] in Waxman model and the degree-based connection method [equation (6)]. Fig. 1 shows the beginning topology created using a random method. The AS number is 149 in 1999.

3.2 Node Addition

After initial topology generation, if a new node is created and is connected to another node, the node addition method is same as above mentioned subsection 3.1: Firstly, the node connectivity rate considering Euclidean distance between two nodes and secondly degree-based connection method using the connectivity rate of the Barabasi-Albert model. In Barabasi-Albert model, a higher degree node presents higher connectivity rate in the existing network [3] [10].

3.3 Node Deletion

Current Internet models only consider the network growing modeling. However, note that a network node can be deleted from social or technical problems. Hence, though the Internet is continuously growing, each network node has its life cycle so that it can be died. In this paper, we derive two node deletion scenarios based on equation (5) and (6). First scenario is based on the distance between two nodes as follows.

$$P_i = \frac{l_i}{\sum l_k} \tag{7}$$

where l_i is the distance value of a node i , and $\sum l_k$ is the distance sum of all nodes.

In distance-based scheme, if a node (i) is far away from a core node and has the maximum distance value l_i , the rate P_i to i is the maximum value. Hence, the node i having the maximum rate value will be deleted. For the second method, it is the case that considers the degree and the distance simultaneously. Hence, even though a node has the maximum distance value, if the degree of node is high, the node can alive. If $\{I_0, I_1, I_2, \dots, I_n\}$ is the distance set of all nodes from a core node, $\{d_0, d_1, d_2, \dots, d_n\}$ is the degree set of all nodes, and the total number of node is m , the rate P_i is given by

$$P_i = \psi \frac{l_i}{\sum_{i \in V} l_i/m} - \delta \frac{d_i}{\sum_{i \in V} d_i/m} \tag{8}$$

where ψ and δ are the shape parameters, $\sum_{i \in V} I_i/m$ and $\sum_{i \in V} d_i/m$ are the average values, and $V = \{0, 1, 2, \dots, n\}$ is the set of all nodes.

Here, we use the average value for each node to make the same effects about the degree and the distance.

3.4 Processing an Isolated Node

After deleting a core node or a node having high degree, the process to isolated neighbor nodes (with zero degree) is required. Hence, it needs to create a new link and

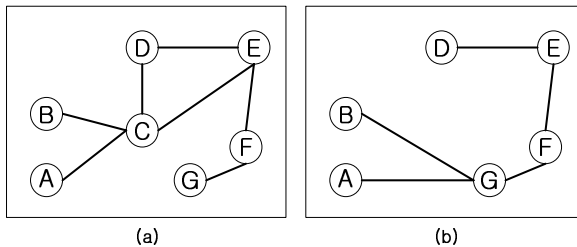


Fig. 2. Processing an isolated node: (a) Before node deletion and (b) After node deletion

connect to another node. As an example, Fig. 2 shows the node isolation process. In Fig. 2(a), the node (*C*) has 4 degrees. If the node (*C*) is deleted, node (*A*) and node (*B*) are isolated. Hence, node (*A*) and node (*B*) should be connected another node [here, node (*G*)]. To create new links, it uses a method similar to that of adding a link when a new node is created: Waxman's node connectivity rate [equation (5)] and the degree-based connection method [equation (6)].

4 Performance Evaluation

We evaluate the correlation between each topology model and the power-law by using three power-law equations (in Section 2.1) [8]. To evaluate the proposed model, we created the network topology in two ways. Table 1 show the current network models without the node deletion function, and Table 2 presents the proposed models that includes the node deletion function and the isolated node process. In the distance-based method, Waxman parameter α and β are 0.6 and 0.4, respectively [see equation (5)].

Table 1. Topology model without a deletion function

Model name	Node addition	
	Node creation	Link connection
A1	Incremental method	Distance-based method
A2	Incremental method	Degree-based method

Table 2. Topology model with a deletion function

Model name	Node addition		Node deletion	Processing to isolated nodes
	Node Creation	Link connection		
B1	Incremental method	Distance-based method	Distance-based method	Distance-based method
B2	Incremental method	Degree-based method	Distance-based method	Distance-based method
B3	Incremental method	Degree-based method	Distance-based method	Degree-based method
B4	Incremental method	Degree-based method	Degree and distance-based method	Distance-based method
B5	Incremental method	Degree-based method	Degree and distance-based method	Degree-based method

In this paper, we used the status of the number of national AS announced by the Korea Network Information Center (KRNIC) [11]. The yearly status of the number of AS to Korea and the world is shown in Table 3. The actually used data in this simulation is from January 1999 to May 2003.

Table 3. AS number from 1999 to 2003

Year	Korea		World	
	AS	Growth rate	AS	Growth rate
1999	149	1.00	11,232	1.00
2000	300	2.01	15,710	1.39
2001	391	2.62	20,128	1.79
2002	433	2.90	23,822	2.12
2003	494	3.31	24,827	2.21

4.1 Evaluation Results

Unfortunately, we can show evaluation results to only the model A2 and the model B5 because of the limited paper space. Fig. 3 and Fig. 4 show graphs that the network topology grows from 149 AS nodes to AS 494 nodes. Hence, to the network growth, the power-law is examined by the growth characteristic of each topology, and results are plotted on the Log-log scale. Fig. 3 is a result using the model (A2) without a node deletion function.

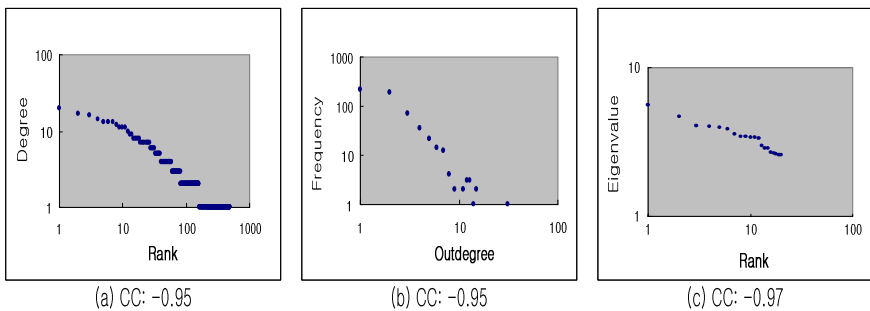


Fig. 3. The network model (A2) : (a) Rank versus degree (b) Degree versus frequency and (c) Rank versus eigenvalue

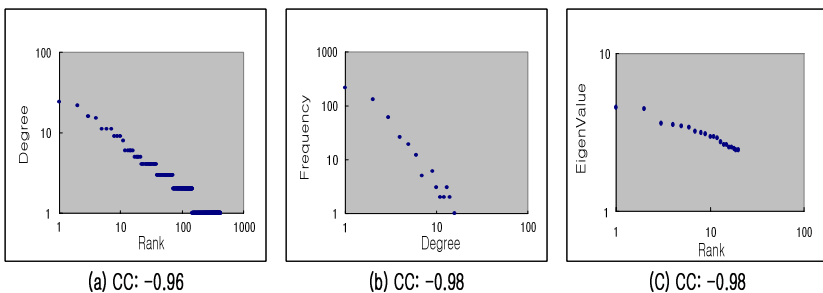


Fig. 4. The network model (B5) : (a) Rank versus degree (b) Degree versus frequency (c) Rank versus eigenvalue

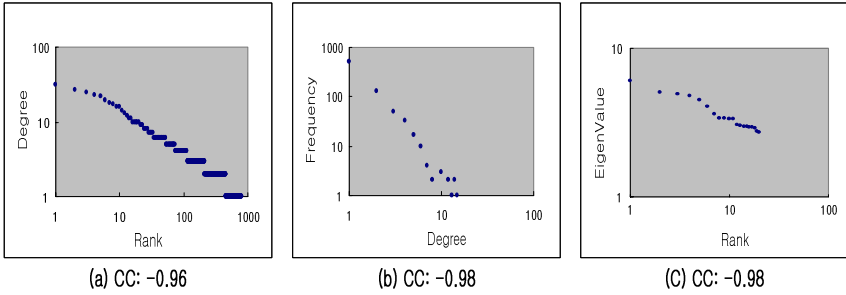


Fig. 5. The predicted network model (B5) after 3 years: (a) Rank versus degree (b) Degree versus frequency and (c) Rank versus eigenvalue

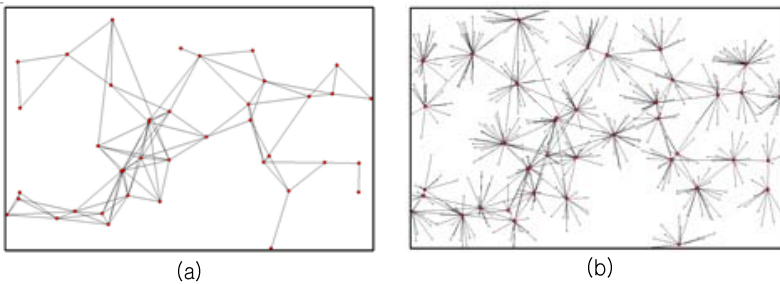


Fig. 6. A network topology example based on the model (B5): (a) core nodes (b) core nodes with leaf nodes

In Fig. 3, three sub-graphs [(a), (b) and (c)] show that each result presents the power-law property because A2 uses a node addition using a gradual increase method and a link creation using the degree-based method. Each CC is -0.95, -0.95, -0.97. Here, CC means the correlation coefficients between the network model and the power-law property. Fig. 4 shows the result of using the model (B5) including a node deletion function and the isolated node process. In Fig. 4, each CC is -0.96, -0.98 and -0.98. The correlation coefficient for the model (B5) show better results of 0.01, 0.03, 0.01 than the ones for the A2 model. Fig. 5 shows the prediction results to the model (B5) after 3 years. We determine the future node number by calculating the average value to all of growth rates in Table 3. After 3 years, the correlation coefficient of B5 is -0.96, -0.98, -0.98. The correlation coefficient of the model (B5) shows better results than that of the model (A2) with 0.01, 0.03 and 0.02. Fig. 6 shows an example of the network topology by the model (B5). The total number of AS node is 490. In Fig. 6(a), the core nodes are firstly generated, and the leaf nodes having the low degree are secondly generated and connected to the core nodes in Fig. 6(b). The proposed model in this paper deals not only with the addition of nodes but it also applies to the node deletion. Hence, we can show more elaborate network model for the Internet evolution.

5 Conclusions

In this paper, we propose an Internet topology model reflects the status of changing topologies caused by the creation and deletion of nodes. The model that we have proposed with the function of node deletion reflects the real Internet topology modeling better compared with current models that simply adds nodes. Furthermore, it shows that the proposed model (B5) has better result values than the model (A2) for predicting the Internet topology in the future. Therefore, the proposed model can better satisfy the power-law that is the characteristic of the Internet compared with current models without the deleting function. We hope the proposed network model that can apply to the topology generator to implement the Internet protocol simulation.

References

1. Zegura, E.W., Calvert, K. L. and Donahoo, M. J., "A Quantitative Comparison of Graph-based Model for Inter-networks," IEEE/ACM Transactions, vol. 5, no. 6, pp.770-783, December 1996.
2. Albert, K. L., Doar, M. B. and Zegura, E. W., "Modeling Internet Topology," IEEE Communications Magazine, Vol. 35, No. 6, pp. 160-163, June 1997.
3. Barabasi, A. L. and Albert, R., "Emergence of Scaling in Random Network," Science, Vol. 286, pp. 509-512, October 1999.
4. Barabasi, A. L., Albert, R. and Jeong, Hawoong, "Scale-free Characteristics of Random Networks: the Topology of the WWW," Physica A, Vol. 28, No. 1, pp. 69-77, 2000.
5. Chen, Q., Chang, H., Govindan, R., Jamin, S., Shenker, S. and Willinger, W., "The Origin of Power Laws in Internet Topologies Revisited," In Proceedings of IEEE INFOCOM 2002, Vol. 2, pp. 23-27, June 2002.
6. Faloutsos, M., Faloutsos, P. and Faloutso, C., "On Power-Law Relationships of the Internet Topology," In Proceedings of ACM Computer Communication Review, pp. 251-261, September 1999.
7. Jin, C., Chen, Q. and Jamin, S., "Inet: Internet Topology Generator," Technical Report CSE-TR-433-00, University of Michigan at Ann Arbor, 2000.
8. Medina, A., Lakhina, A., Matta, I. and Byers, J., "BRITE: An Approach to Universal Topology Generation," In Proceeding of MASCOTS 2001, pp. 346-353, August 2001.
9. Medina, A., Matta, I. and Byers, J., "On the Origin of Power Laws in Internet Topologies," Computer Communication Review, Vol. 30, No. 2, pp. 18-28, April 2000.
10. Waxman, B. M., "Routing of Multipoint Connections," IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, pp. 1617-1622, December 1998.
11. <http://www.krnic.or.kr>.

Attack Strategies on Complex Networks

Lazaros K. Gallos¹, Reuven Cohen², Fredrik Liljeros³, Panos Argyrakis¹,
Armin Bunde⁴, and Shlomo Havlin⁵

¹ Department of Physics, University of Thessaloniki, 54124 Thessaloniki, Greece
gallos@physics.auth.gr

² Department of Physics, Bar-Ilan University, 52900 Ramat-Gan, Israel

³ Department of Sociology, Stockholm University 106 91 Stockholm, Sweden

⁴ Institut für Theoretische Physik III, Justus-Liebig-Universität Giessen
Heinrich-Buff-Ring 16, 35392 Giessen, Germany

Abstract. In this work, we estimate the resilience of scale-free networks on a number of different attack methods. We study a number of different cases, where we assume that a small amount of knowledge on the network structure is available, or can be approximately estimated. We also present a class of real-life networks that prove to be very resilient on intentional attacks, or equivalently much more difficult to immunize completely than most model scale-free networks.

1 Introduction

A large number of diverse systems in society, nature and technology can be described by the concept of a network [1, 2]. In a network the form of inter-relations between the system parts determines many structural and dynamic properties of the system. One such property that has received considerable attention is the robustness of a network under failures [3, 4] or intentional attack [3, 5, 6]. Equivalently, from a sociological point of view, the robustness of a network can be related to an immunization process, where immunized nodes no longer transmit a disease, and thus the destruction of a spanning cluster in the network means that the population is immune to a disease, which will soon die out because it will encounter non-susceptible nodes. In such cases we are mainly interested in using the lowest possible number of vaccinations, either for reasons of increased cost or unavailability of a large number of vaccines. These strategies strongly influence the form of the resulting network, which in turn affects important dynamic properties, such as delivery time in the Internet, delays in information or virus spreading, etc [7].

In the course of an intentional attack nodes of the network are removed in decreasing order of their degree (number of connections to other nodes). This is considered to be the most harmful type of attack on a network, since the removal of the hubs results in the largest possible damage. This removal process has many and important implications, since depending on the application, it may describe the resilience of a network, such as the Internet, or the required number of vaccinations for immunization considerations, etc. For a scale-free network, where the probability that a node has a given number of links decays

as a power-law $P(k) \sim k^{-\gamma}$, it has been shown that the critical percentage f_c of removed nodes that results in network desintegration is very low (less than $f_c = 0.07$) [5, 6]. It is, thus, a well-established fact, supported by analytic results and simulations on model and real-life networks, that a scale-free network is very vulnerable to intentional attacks (where f_c is close to 0), although the same network is extremely robust under random node failures (where $f_c \simeq 1$) [4].

2 Attacks with Limited Network Knowledge

In many cases, it is possible that the robustness of a node depends on its connectivity, i.e. the probability of damaging a node either by failure or by an external attack depends on the degree k of the node. We simulate this situation by using the probability $W(k) \sim k^\alpha$ for a node with degree k to become inactive. The parameter α can be regarded as a measure of our knowledge on the network structure. When $\alpha < 0$ nodes with low degree are more vulnerable, while for $\alpha > 0$ high-degree nodes are removed with higher probability than the low degree nodes. The cases $\alpha = 0$ and $\alpha \rightarrow \infty$ represent the known random removal and targeted intentional attack, respectively.

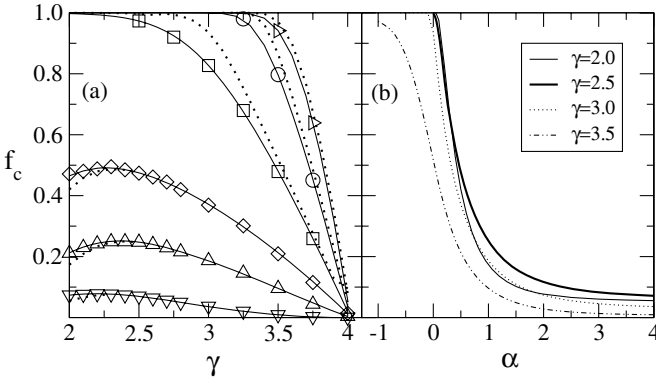


Fig. 1. (a) Values of f_c vs γ for different α values: (bottom to top) $\alpha = 4, 1, 0.5, 0, -0.5, -1$. Symbols represent simulation data ($N = 10^6$ nodes) from 100–300 different network realizations. Solid lines are the theoretical predictions for finite-size networks, while dashed lines correspond to infinite-size networks. (b) Values of f_c vs α for infinite-size networks and different γ values.

Results of simulations on model networks for the critical fraction f_c of the nodes that need to be removed before the destruction of the spanning cluster are presented in Fig. 1. A theoretical result for this problem has been presented in [8], which can be seen in the figure to be in excellent agreement with the simulations. For $\gamma < 3$, f_c becomes smaller than 1 already for very small positive α values, and decays rapidly with increasing α . Accordingly, by a very small preference probability to remove highly connected nodes, which arises, for example,

in an intentional attack with very little knowledge of the network structure, this network can be destroyed by removing a comparatively small fraction of nodes. Above $\alpha = \gamma - 1$, f_c saturates, which means that the knowledge available to the attacker in this case is sufficient to destroy the network most efficiently.

Our results show that little knowledge on the highly connected nodes in an intentional attack reduces the threshold drastically compared to the random case. Thus, a large network can be damaged efficiently even when only a small fraction of hubs is known to the attacker. For immunization of populations this means that if we are able to identify (and immunize) even with small probability the virus spreaders we can significantly reduce the spreading threshold.

3 Acquaintance Immunization with Limited Knowledge

Another method of attacking the system, with application in immunization strategies, is the acquaintance immunization method [9]. According to this scheme a random node is selected and then points to a random acquaintance along one of its links. The node at the other end of the link is the one to be immunized (removed). This method achieves great efficiency in lowering the percolation threshold. Here, we consider the probability of further improving the efficiency of this process, provided that we use partial knowledge on the network structure.

We select a percentage p of a network comprising N nodes and ask them to direct us to a random acquaintance of theirs, which will then be immunized. The initial selection of the pN nodes is based on partial information on their connectivity, so that a node i with k_i links has a probability $W(k_i) \sim k_i^\alpha$ of being approached (the parameter α has the same meaning as in the previous section). We increase the value of p up to a value p_c where the fraction f_c of actually immunized nodes results to the arrest of the epidemic (or equivalently to the destruction of the spanning cluster).

We present a theoretical analysis of the problem, based on the arguments presented in Ref. [9]. We consider a network where a fraction p of its nodes have been selected and have pointed to a fraction f of unique nodes that have been immunized. We assume that in the network there are no degree-degree correlations, so that the probability of a node with k links to be connected to a node with k' links is independent of k , $\phi(k') \equiv p(k'|k) = k'P(k')/\langle k \rangle$.

For our derivation we assume that loops can be neglected and the nodes are located on layers, l , from a chosen origin. We denote the number of nodes with degree k in the layer l by $n_l(k)$. The event of a node with degree k being susceptible (not immunized) is denoted by s_k . As a starting point for the calculation of $n_l(k)$ we use Eq. (1) from Ref. [9]

$$n_{l+1}(k') = \sum_{k=1}^{k_{\max}} n_l(k)(k-1)p(k'|k, s_k)p(s_{k'}|k', k, s_k). \tag{1}$$

The upper value k_{\max} is taken equal to the natural cutoff $k_{\max} = N^{1/(\gamma-1)}$, while $p(k'|k, s_k)$ denotes the probability of reaching a node of degree k' by following a

link from a susceptible node with degree k and $p(s_{k'}|k', k, s_k)$ is the probability that this k' -degree node is also susceptible.

A random node of degree k is selected with probability $k^\alpha/(N\langle k^\alpha \rangle)$ where $\langle k^\alpha \rangle = \sum k^\alpha P(k)$. In order to find the probability for a random acquaintance to be immunized we divide by $1/k$, so that the probability of immunizing this specific acquaintance is $k^{\alpha-1}/(N\langle k^\alpha \rangle)$. The probability that this node is not selected is $1 - k^{\alpha-1}/(N\langle k^\alpha \rangle)$ and after Np immunization attempts it becomes

$$\nu_p(k) = \left(1 - \frac{k^{\alpha-1}}{N\langle k^\alpha \rangle}\right)^{Np} \sim \exp\left(-\frac{k^{\alpha-1}}{\langle k^\alpha \rangle}p\right). \quad (2)$$

Since the network is uncorrelated we consider the average value $\nu_p = \langle \nu_p(k) \rangle = \sum_k \nu_p(k)\phi(k)$, so that the probability that a node with degree k is susceptible is, in general, $p(s_k|k) = \nu_p^k$. If the degree of one neighbor is known to be k' this probability becomes $p(s_k|k, k') = \nu_p^{k-1} \exp\left(-\frac{k'^{\alpha-1}}{\langle k^\alpha \rangle}p\right)$. Since immunization of a node is independent of the probability that its neighbor is also immunized we also have $p(s_k|k, k') = p(s_k|k, k', s_{k'})$. Combining the above results with the Bayes rule and Eq. (2) we finally get the expression:

$$n_{l+1}(k') = n_l(k') \sum_{k=1}^{k_{\max}} \phi(k) \nu_p^{k-2} (k-1) \exp\left(-\frac{2k^{\alpha-1}}{\langle k^\alpha \rangle}p\right). \quad (3)$$

When the sum in the above expression is greater than 1 then the number of susceptible nodes increases with increasing layer index l . When the sum is less than 1 the percolation phase disappears. At the critical concentration p_c this sum is, thus, equal to 1, i.e.

$$\sum_{k=1}^{k_{\max}} \phi(k) \nu_{p_c}^{k-2} (k-1) \exp\left(-\frac{2k^{\alpha-1}}{\langle k^\alpha \rangle}p\right) = 1. \quad (4)$$

Next, we obtain the critical value p_c by numerically solving this equation and we compute the critical fraction of immunized nodes, i.e. the fraction of not susceptible nodes:

$$f_c = 1 - \sum_{k=1}^{k_{\max}} P(k)p(s_k|k) = 1 - \sum_{k=1}^{k_{\max}} P(k)\nu_{p_c}^k. \quad (5)$$

The numerical solution of Eq. (5) for f_c as a function of α for networks with different γ exponents is compared in Fig. 2 with simulation data.

For networks with $\gamma < 3$ the critical threshold is minimized at values of $\alpha \simeq 1$, which is the optimum value for the presented strategy, while for $\gamma \geq 3$ the threshold presents a different behavior and decreases monotonically with increasing α values. In practice, the process at $\alpha = 1$ is equivalent to selecting a random link and immunizing one of the two nodes attached to the given link (provided the uncorrelated network hypothesis holds). It is also interesting to notice that

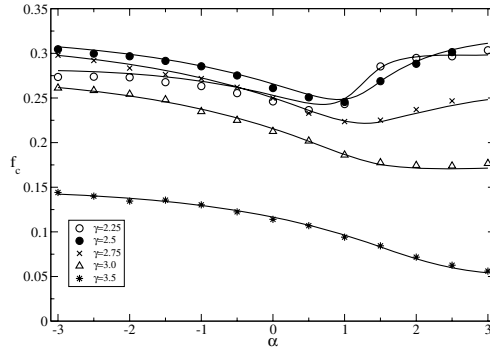


Fig. 2. Critical immunized fraction f_c of the population as a function of α for random scale-free networks with different γ exponents (shown in the plot). Network size is $N = 10^5$ nodes. Symbols are the results of simulations and lines represent the analytic solution (Eqs. 4 and 5).

up to the value $\alpha = 1$ the acquaintance immunization strategy is superior to direct immunization of the initially selected nodes, but close to this value the two methods yield a similar value for f_c . When $\alpha > 1$ the direct immunization method becomes more efficient than the acquaintance immunization strategy.

4 Robust Real-Life Scale-Free Networks

In this section we show that there exists a large class of networks, that are usually found in nature and society and have already been characterized as scale-free, but nevertheless remain robust against removal of the most connected nodes. We first present the results for real-life networks and then introduce a modified version of scale-free networks, for which our analytic and simulation treatment support these findings.

To demonstrate this issue we performed intentional attacks and random nodes removal to many different real-life networks. Although many of these systems behave in a similar way to the model network (where f_c is usually less than 10%) there is a number of networks, such as actors collaboration and science citations, where the intentional attack requires removal of a considerable portion of the network nodes, which is of the order of 65%. In order to outline the common feature of these networks, in Fig. 3 we present the degree distribution of these networks. These distributions have a flat or rising part at low-degree nodes and only after a threshold value the distribution decays as a power-law.

We use a general model for simulating similar networks. We consider networks whose degree distribution is uniform up to a threshold value k_c and for larger values decays as a power law $k^{-\gamma}$. The exact form of the distribution (plotted also for $k_c = 50$ and $\gamma = 2.5$ in Fig. 3) is

$$P(k) = \begin{cases} \frac{\gamma-1}{\gamma} k_c^{-1} & 1 < k < k_c \\ \frac{\gamma-1}{\gamma} k_c^{\gamma-1} k^{-\gamma} & k > k_c \end{cases} . \tag{6}$$

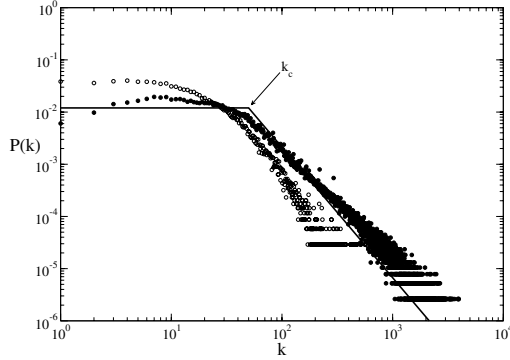


Fig. 3. Degree distributions for IMDB actors (filled symbols) and HEP citations (open symbols). The solid line represents a typical degree distribution (Eq. 6) that we used as a model.

We calculate the critical threshold f_c for such a network based on ideas introduced by Cohen et al [5] and Dorogovtsev and Mendes [11]. Nodes are removed according to their initial degree. An intentional attack results in the disruption of the network. We consider that the degrees of the nodes for the resulting network are given by the parameter \tilde{k} , with corresponding averages

$$\langle \tilde{k} \rangle = \int_1^{\tilde{K}} kP(k)dk, \quad \langle \tilde{k}^2 \rangle = \int_1^{\tilde{K}} k^2P(k)dk. \quad (7)$$

The effect of an intentional attack is to remove all nodes of a network whose degree is larger than a cutoff value \tilde{K} , i.e. $\tilde{k} \in [1, \tilde{K}]$. This also implies that f_c equals

$$f_c = 1 - \int_{\tilde{K}}^{\infty} P(k)dk. \quad (8)$$

At the same time, removal of a node leads to removing all its links to other nodes. We consider random networks with no correlations in the nodes connections, which means that a removal of a node results in removal of random links with probability

$$\tilde{p} = \frac{\int_{\tilde{K}}^{\infty} kP(k)dk}{\int_1^{\infty} kP(k)dk} = 1 - \frac{\langle \tilde{k} \rangle}{\langle k \rangle}. \quad (9)$$

It has been shown [4, 10] that a random network loses its large-scale connectivity after the removal of a critical fraction f_c of nodes, which behaves as

$$f_c = 1 - \frac{1}{\kappa - 1} \quad (10)$$

where $\kappa \equiv \langle k^2 \rangle / \langle k \rangle$ as usual. We use the above equation for the network resulting after the attack, i.e. we substitute f_c with \tilde{p} from Eq. 9 and $\kappa = \langle \tilde{k}^2 \rangle / \langle \tilde{k} \rangle$. After a few trivial steps Eq. 10 becomes

$$\langle \tilde{k}^2 \rangle - \langle \tilde{k} \rangle = \langle k \rangle. \quad (11)$$

This formula, which is exact, has been already proved in Refs. [5, 11].

In order to use Eq. 11 we need to know whether the value of \tilde{K} is larger or smaller than the threshold value of the distribution k_c . We have considered each case separately, but when $\tilde{K} > k_c$ there is no solution to the problem (which is also verified by our simulations where always $\tilde{K} < k_c$). Calculation of the involved integrals yields

$$\langle \tilde{k} \rangle \simeq \frac{\gamma - 1}{2\gamma} \frac{\tilde{K}^2}{k_c}, \tag{12}$$

and

$$\langle \tilde{k}^2 \rangle \simeq \frac{\gamma - 1}{3\gamma} \frac{\tilde{K}^3}{k_c}. \tag{13}$$

The average value of the initial degree distribution $P(k)$ (Eq. 6) can be approximated with the assumption that $k_{\max} = \infty$. However, for low γ values this assumption does not work well and we can compute the integral up to the maximum value $k_{\max} = K$, which can be computed from the relation $\int_{k_{\max}}^{\infty} P(k) = 1/N$, and is given in our case by $K = k_c N^{1/(\gamma-1)} \gamma^{1/(1-\gamma)}$. This results in a correction $x = 2N^{(2-\gamma)/(\gamma-1)} \gamma^{1/(1-\gamma)}$ to the average value of the distribution, which finally becomes

$$\langle k \rangle = \frac{(\gamma - 1)k_c}{2(\gamma - 2)}(1 - x). \tag{14}$$

Combining Eqs. 11-14 we get

$$2\tilde{K}^3 - 3\tilde{K}^2 = \frac{3\gamma k_c^2}{\gamma - 2}(1 - x). \tag{15}$$

We can now compute the value of \tilde{K} from Eq. 15 and substitute it to Eq. 8, which can also be written as

$$f_c \simeq 1 - \frac{\gamma - 1}{\gamma} \frac{\tilde{K}}{k_c}. \tag{16}$$

The numerical solution of Eqs. 15 and 16 is shown in Fig. 4 as a function of γ for different values of the threshold value k_c . In the same figure we also plot results of simulations on model networks.

Comparison of the curves in Fig. 4 to the intentional attack on regular scale-free networks shows a dramatic increase in the value of f_c , over the entire γ range. Increase of the threshold value k_c enhances the stability of the network. For $k_c = 10$ the critical fraction is already above 40%, while when $k_c = 100$ the value of f_c lies in the range 70-80%. This, of course, is also a consequence of the increasing mean degree of nodes as we increase k_c , that makes all nodes in the system better connected.

These findings provide a structure that is very robust against both random failures and targeted attacks. This optimization is desirable in most cases. Such a structure, which we have seen in many instances emerges naturally, may be used to efficiently protect a network against most attacks.

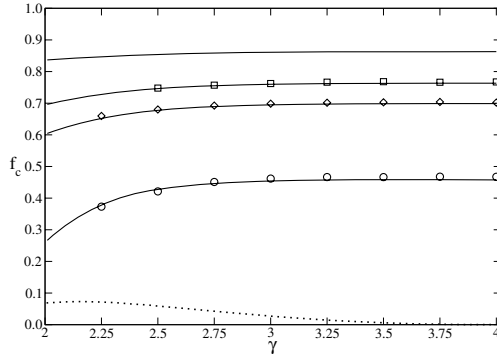


Fig. 4. Critical fraction f_c of removed nodes for networks that undergo an intentional attack, as a function of the exponent γ . From top to bottom: $k_c = 500, 100, 50$, and 10 . Solid lines represent the numerical solution of Eqs. 15 and 16, while symbols are simulation results. The dashed curve corresponds to pure scale-free networks (Ref. [5]).

Acknowledgement

This work was supported by a European research NEST/PATHFINDER project DYSONET 012911.

References

1. R. Albert and A.-L. Barabasi, *Rev. Mod. Phys.* **74**, 47 (2002).
2. S.N. Dorogovtsev and J.F.F. Mendes, *Adv. Phys.* **51**, 1079 (2002).
3. R. Albert, H. Jeong, and A.L. Barabási, *Nature (London)* **406**, 378 (2000).
4. R. Cohen et al., *Phys. Rev. Lett.* **85**, 4626 (2000).
5. R. Cohen et al., *Phys. Rev. Lett.* **86**, 3682 (2001).
6. D.S. Callaway et al., *Phys. Rev. Lett.* **85**, 5468 (2000).
7. D.J. Watts, *Proc. Nat. Ac. Sci.* **99**, 5766 (2002).
8. L.K. Gallos et al., *Phys. Rev. Lett.* **94**, 188701 (2005).
9. R. Cohen, S. Havlin, and D. ben-Avraham, *Phys. Rev. Lett.* **91**, 247901 (2003).
10. G. Paul, S. Sreenivasan, and H.E. Stanley, preprint arxiv:cond-mat/0507202 (2005).
11. S.N. Dorogovtsev and J.F.F. Mendes, *Phys. Rev. Lett.* **87**, 219801 (2001).

Elementary Modules in Games Networks

Matthieu Manceny and Franck Delaplace

IBISC, FRE 2873 CNRS - University of Evry
523 Place des Terrasses, 91000 Evry, France

Abstract. In this paper we propose an original modular extension of game theory named *games network*. The objective of games networks is to provide a theoretical framework which suits to modular dynamics resulting from different local interactions between various agents and which enables us to describe complex system in a modular way. Games networks describes situations where an agent can be involved in several different games, with several different other agents, at the same time. In particular, we focus on the determination of *global equilibria*, resulting from the composition of local equilibria for each game of the network.

However, several games networks can represent the same dynamics. We define the notion of dependence between agents, which allows us to compute a *games network normal form*. This normal form emphasizes the *elementary modules* which compose the games network.

Keywords: complex systems, modularity, game theory, networks, dynamics.

1 Introduction

Analysis of complex systems is often based on the studies of relationships between components instead of elements themselves. This puts the emphasis on the way to analyze interactions. From modeling standpoints, networks provide a suitable framework to describe interactions (edges) of components (vertices). With networks, the description remains static and it is mainly focused on the structural analysis of the properties of the system. For example, a network where the connectivity degree follows a power law identifies robustness properties to the “not targeted” attacks ([1, 3, 6]). Thus, the study of interacting networks within a modeling framework is based on a parallel between structural properties of the network (e.g. power law) and dynamical properties (e.g. robustness).

In order to improve the framework by including dynamical aspects for the analysis of interactions, we propose to mix two formalisms: network formalism and game theory.

Game theory has been pioneered by von Neumann and Morgenstern to define a theoretical framework to model complex interactions between agents. It studies how the interacting agents (or *players*) make their choices (or *strategies*) evolved considering their interactions with other players ([13]). Applications of game theory are larger than “Games”, and characterize complex interactions in fields such as Biology ([8, 11]), Economy ([7, 9]) or Computer Science ([2, 12]). The

choices players make characterize the dynamical aspects of game theory. The notion of *Nash equilibrium* captures the steady states of a game.

Theory of games networks is an extension of game theory where a player can be involved in several games simultaneously. Games networks can be viewed as a “network of games and players” where players are connected to the games they participate to. With games networks, we describe the interactions as a set of modular activities where each game represents a module of interactions. We can study how the dynamics of one game, defined by *local equilibria*, influences the dynamics of the whole network, defined by *global equilibria*.

In order to analyze and understand interactions between components of a network, we search for the elementary interactions within this network. We define a separation algorithm which decomposes one game in its elementary modules underlining elementary interactions. From their reduced size, elementary modules should be more comprehensible than the games of the starting network. Moreover, elementary modules identify structures which would be impossible to characterize if agents are separately considered.

The paper is organized as follow: section 2 presents fundamental notions of game theory and theory of games networks; section 3 deals with research of elementary modules and describes the separation algorithm. We conclude in section 4.

2 Theory of Games Networks

In this section we present an original extension of game theory: the theory of games networks. In order to ease the reading of the paper, we first briefly recall notions of game theory; then we introduce the extension. The reader may refer to [10] or [5] for a more complete presentation.

2.1 Game Theory

Strategic Games. Strategic game theory proposes a model of interactions where interacting agents, the *players*, choose their action, their *strategy*, once and for all and simultaneously. Moreover, each player is rational — it aims at maximizing its *payoff* — and *perfectly informed* of other players’ payoffs. The formal definition of a strategic game is as follow:

Definition 1 (Strategic game). *A strategic game is a triplet $\langle A, C, u \rangle$ where:*

- *A is the set of players.*
- *$C = \{C_i\}_{i \in A}$ is a set of strategy sets; $C_i = \{c_i^1, \dots, c_i^{m_i}\}$ is the set of player i ’s strategies.*
- *$u = (u_i)_{i \in A}$ is the payoff vector; $u_i : \times_{i \in A} C_i \mapsto \mathbb{R}$ is a function which maps a payoff for player i considering a game configuration, i.e. other players’ strategies.*

Payoff table	Nash equilibria									
<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">x/y</td> <td style="padding: 5px;">Off</td> <td style="padding: 5px;">On</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">Off</td> <td style="padding: 5px;">(1, 1)</td> <td style="padding: 5px;">(0, 2)</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">On</td> <td style="padding: 5px;">(3, 0)</td> <td style="padding: 5px;">(-1, -1)</td> </tr> </table>	x/y	Off	On	Off	(1, 1)	(0, 2)	On	(3, 0)	(-1, -1)	$\{(On, Off), (Off, On)\}$
x/y	Off	On								
Off	(1, 1)	(0, 2)								
On	(3, 0)	(-1, -1)								

Fig. 1. Example of a 2×2 strategic game

Representation by Table. 2×2 strategic games — 2 players with 2 strategies — are usually used to present game theory notions. Such a game can be represented by a table where first player’s strategies are in line, and second player’s ones in column. Considering example from fig. 1: if player x play its Off strategy y its On strategy, then the payoff attributed to x is 0 and 2 for player y .

Nash Equilibrium. Nash equilibrium is a central concept which captures the steady configurations of a strategic game:

Definition 2 (Nash equilibrium). Let $\langle A, C = \{C_i\}_{i \in A}, u = (u_i)_{i \in A} \rangle$ be a strategic game. A Nash equilibrium is a game configuration $c^* \in \times_{i \in A} C_i$ such that:

$$\forall i \in A, \forall c_i \in C_i, u_i((c_{-i}^*, c_i)) \leq u_i(c^*)$$

with (c_{-i}^*, c_i) equivalent to the game configuration c^* but where player i plays its strategy c_i rather than c_i^* .

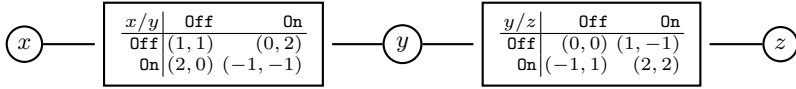
In a Nash equilibrium, the strategy played by player i is the best possible response to strategies played by other players. In other words, no agent can unilaterally deviate from a Nash equilibrium without decreasing its payoff. Considering example from fig. 1, 2 Nash equilibria exist: $(x = Off, y = On)$ and $(x = On, y = Off)$.

2.2 Games Networks

Strategic Games Networks. In game theory, all the agents are interacting together. Theory of games networks is an original framework which extends game theory and allows a modular description of the network dynamics. Thus, players can participate to several games simultaneously. Games which composed the games network can be seen as dynamical modules which describe the local interactions between agents. The formal definition of a strategic games network is as follow:

Definition 3 (Games network). A strategic games network is a triplet $\langle A, C, \mathcal{U} \rangle$ where:

- A is the set of players.
- $C = \{C_i\}_{i \in A}$ is a set of strategy sets; $C_i = \{c_i^1, \dots, c_i^{m_i}\}$ is the set of player i ’s strategies.
- $\mathcal{U} = \{\langle A_j, u^j \rangle\}$ is a set of games. For each game, $A_j \subseteq A$ is the set of players and $u^j = (u_i^j : \times_{i \in A_j} C_i \mapsto \mathbb{R})_{i \in A_j}$ is the payoff vector.



Global equilibria: $\{(x = \text{On}, y = \text{Off}, z = \text{Off}), (x = \text{Off}, y = \text{On}, z = \text{On})\}$

Fig. 2. Example of a 3-players-2-games games network

Given a game, it is not necessary to recall the strategies available for an agent because they are identical for all the game this agent participates to. Thus, strategies are associated to the agent rather than to each game.

Graphical Representation. Games networks are represented by bipartite graphs (fig. 2). In such a graph, players are represented by a circle and games by rectangles. Players are connected to games they participate to.

Equilibria. Two types of dynamics emerge from the games network representation: the first one is local to each game and the second one is global to the whole network. Thus, two notions of equilibria have been defined: local equilibria and global equilibria. Local equilibria correspond to Nash equilibria for each game which composed the network. Considering example from fig. 2: two local equilibria exist for the x/y game ($(x = \text{Off}, y = \text{On})$ and $(x = \text{On}, y = \text{Off})$) and two local equilibria for the y/z game ($(y = \text{Off}, z = \text{Off})$ and $(y = \text{On}, z = \text{On})$). Global equilibria correspond to a game configuration which define local equilibria for all the games of the network. Global equilibria can be computed by a combination of local equilibria. Considering example from fig. 2: strategies available for player y are Off or On, which corresponds to two global equilibria, $(x = \text{On}, y = \text{Off}, z = \text{Off})$ and $(x = \text{Off}, y = \text{On}, z = \text{On})$.

3 Elementary Modules

3.1 Structure and Network Equivalence

In games networks, each game of the network is naturally identified to a module. Thus, a games network can be seen as a composition of modules linked through the agents. Each module defines a *local dynamics* and the network structure — the way the modules are linked — defines a *global dynamics*. These dynamics are observed by their steady states: local and global equilibria.

However, a same dynamics can be modeled by different structures. Considering the example from fig. 3, the one-game-three-players network, on the left, has the same global equilibria, and the same dynamics, than the two-games network, on the right. We say the two networks are *equivalent*.

Thus, we have to search for a “*normal form*”, that is a canonical representation of a games networks. A games network normal form is defined as an equivalent games network whose games involved as few players as possible. Games in a games network normal form are called “*elementary modules*”.

3.2 Algorithm

Algorithm from fig. 4 separates a game in elementary modules¹. It is based on the notion of player dependence.

Dependence. Intuitively, a player a depends on a player b if a 's payoffs are altered by b 's strategies. Formally, dependence is defined as follow:

Definition 4 (Dependence). Let $\langle A, C, u \rangle$ be a strategic game. Let $j \neq i \in A^2$ be two agents. j depends on i , denote by $i\delta_u j$, if:

$$\exists c_i \in C_i, \exists c'_i \in C_i, \exists c_{-i} \in C_{-i}, u_j(c_{-i}, c_i) \neq u_j(c_{-i}, c'_i)$$

More precisely, the algorithm refers to the notion of predecessors:

Definition 5 (Predecessors). Let $\langle A, C, u \rangle$ a strategic game. We note $\delta_u^-(j)$, $j \in A$, the set of j 's predecessor:

$$\forall j \in A, \delta_u^-(j) = \{i \in A | i\delta_u j \wedge i \neq j\}$$

Notions of dependence and predecessors are used to underline interactions between agents. It allows us to determine which agents have to be involved in the same elementary module.

Payoffs. Once we have find players involved in an elementary module, payoffs have to be attributed. Let $a \in A$ be a player which participates to an elementary module G :

- if all of a 's predecessors are in G , we can easily compute a 's payoffs, because none of the absent players has any influence on a 's payoffs. The *pick* function in the separation algorithm chooses one configuration of the starting game where strategies for a 's predecessors are identical to their strategies from the elementary module, and compute a 's payoffs.
- if one of a 's predecessors is not in g , then a 's payoffs are 0.

The separation is illustrated in fig. 3; fig. 4 details the algorithm.

4 Conclusion

In this paper, we presented the theory of games networks to study modularity of interacting networks. Theory of games networks extends game theory with the possibility to define local interactions between agents. These local interactions are modeled by the different games which constitute the network, and are observed through their local equilibria. At the scale of the whole network, local equilibria combine to form global equilibria. The same complex dynamics can be represented by several different games networks. We were interested in a

¹ For more information, please refer to [5].

Game (payoffs table)

x	y	z	u^x	u^y	u^z
Off	Off	Off	1	2	-1
Off	Off	On	1	2	0
Off	On	Off	-1	0	-1
Off	On	On	-1	0	0
On	Off	Off	0	0	0
On	Off	On	0	0	-1
On	On	Off	3	3	0
On	On	On	3	3	-1

Normal form

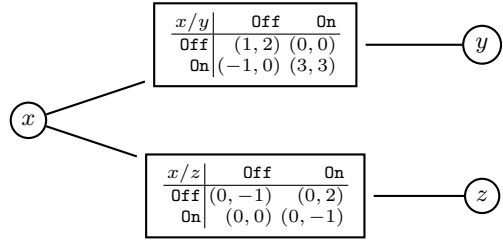


Fig. 3. A 3-players game and its normal form

```

function Separate( $\langle A, C, u \rangle$  : a game)
   $U' := \emptyset$ ;  $g := 0$ ;
  /*Computation of the number of games to be created*/
  For all  $i \in A$ 
     $g := g + 1$ ;
     $\mathbf{agent}(g) := i \cup \delta_u^-(i)$  ;
  EndForAll
   $U = [1 : g]$ ;
  For all  $g' \in [1 : g]$ 
     $U := U - \{g'' \in U | \mathbf{agent}(g'') \subset \mathbf{agent}(g') \vee (\mathbf{agent}(g') = \mathbf{agent}(g'') \wedge g'' < g')\}$ ;
  EndForAll
  /*Attribution of payoffs*/
  For all  $g \in U$ 
    For all  $j \in \mathbf{agent}(g)$ 
      If  $\delta_u^-(j) \cap \mathbf{agent}(g) = \delta_u^-(j)$  Then
        For all  $c \in \times_{i \in \mathbf{agent}(g)} C_i$ 
           $u_j^g(c) := \mathbf{pick}(c, j)$ 
        EndForAll
      Else
        For all  $c \in \times_{i \in \mathbf{agent}(g)} C_i$ 
           $u_j^g(c) := 0$ 
        EndForAll
      EndIf
    EndForAll
  EndForAll
   $U' = U' \cup \{\langle \mathbf{agent}(g), u^g \rangle\}$ ;
EndForAll
return  $\langle A, C, U' \rangle$ ;

```

Fig. 4. Separation algorithm

canonical representation of games networks — the normal form — where each game involved as less agents as possible. In normal form, games are qualified as elementary modules and connect the “more connected” agents.

Theory of games networks has been used in a real case to model a part of the Plaminogen Activator system (PAs) which is involved in the migration of cancer cells. The games network of PAs is composed of 10 biological agents

(such as molecules or proteins) and 6 games. The modelisation has underlined the central role of one player (the Plasminogen Activator Inhibitor, PAI-1), and the existence of two global equilibria which correspond to physiological states of the cells. These results were found in experiments. The reader may refer to [4] to know more about PAs and its games networks modelisation.

References

1. R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
2. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, Florida, October 1997.
3. A. Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means*. Plume, 2003.
4. C. Chettaoui, F. Delaplace, M. Manceny, and M. Malo. Games Network & Application to PAs system. In *Information Processing in Cells and Tissues (IPCAT)*, 2005.
5. F. Delaplace and M. Manceny. Games network. Technical Report 101-2004, Laboratoire de Méthodes Informatiques (LaMI), CNRS-UMR 8042, University of Evry, 2004. <http://www.lami.univ-evry.fr/~mmanceny/>.
6. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.
7. D. M. Kreps. *A Course in Microeconomic Theory*. Princeton University Press, 1990.
8. J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge Univ. Press, 1982.
9. R. D. McKelvey and A. McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1, pages 87–142. Elsevier, 1996. <http://econweb.tamu.edu/gambit/>.
10. R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
11. M. A. Nowak and K. Sigmund. Evolutionary dynamics of biological games. *Sciences*, 303(6):793–799, februar 2004.
12. C. H. Papadimitriou. Game theory and mathematical economics: a theoretical computer scientist’s introduction. In *42nd IEEE Symposium on Foundations of Computer Science: Proceedings*, pages 4–8, 2001.
13. J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, Princeton, New Jersey, first edition, 1944.

A New Analysis Method for Complex Network Based on Dynamics of Spin Diffusion

Makoto Uchida¹ and Susumu Shirayama²

¹ Graduate School of Engineering, the University of Tokyo,
5-1-5 Kashiwanoha, Kashiwa city, Chiba, 277-8568, Japan
uchida@race.u-tokyo.ac.jp

² Research into Artifacts, Center for Engineering, the University of Tokyo,
5-1-5 Kashiwanoha, Kashiwa city, Chiba, 277-8568, Japan
sirayama@race.u-tokyo.ac.jp

Abstract. We propose a new analysis method for a complex network based on a simple spin diffusion model. The model is constructed by a local interaction between vertices, as is in the spin dynamics. Several numerical experiments on network models are performed systematically under various initial conditions. The results strongly depend on the network structures, also on the initial conditions, while most conventional measures of the networks are almost same. It may be shown that the difference of dynamics induced by such interaction reveals a new characteristic feature of the network structure.

1 Introduction

Complex network analysis methods have been utilized to inquire complicated structures and functions. Early works had shown that many real-world networks in various fields have the common structural properties, typically such as 'small-world', 'scale-free' or community structure [1, 2]. Recently, various methods or new measures for analyzing a unique structure of networks are being proposed. Other works have paid attention to phenomena on networks; percolation, epidemic spreading, traffic on networks, *etc* [2]. A fundamental problem is to understand how the topological structure affects the dynamics that it supports [3].

In this paper, we propose a simple model of diffusion on networks, based on a local interaction between vertices, then apply this model on several models of network numerically. The resulting phenomena are very different depending on both initial conditions and the structure of networks itself. The difference of the diffusion dynamics by our method may reveal a new characteristic of the network structure.

2 Proposed Method

2.1 Diffusion Model

We consider a simple diffusion process on networks. First, a variable $\sigma_i(t)$ which takes +1 or -1 is defined. This variable represents a certain state of a vertex v_i at a time step t . Then, we set $\sigma_i(0)$ to all vertices as an initial condition in some manners. In

the consequent section, we explain the details of the initial condition. The state at the subsequent step $t + 1$ is determined by the present state of vertices to which an edge from v_i exists. There is an interaction only among neighboring vertices. $\sigma_i(t + 1)$ is determined by the following rule:

$$\sigma_i(t + 1) = \text{sgn} \left\{ \sum_j^N a_{ij} \sigma_j(t) \right\} \quad (1)$$

$$\sigma_i(t + 1) = \sigma_i(t), \quad \text{if} \quad \sum_j^N a_{ij} \sigma_j(t) = 0 \quad (2)$$

where $a_{ij} = a_{ji}$ is the i - j (j - i) element of an adjacency matrix, which takes the value of 1 if there is an edge between vertices v_i and v_j , otherwise 0. The states of vertices are updated synchronously at each step.

2.2 Initial Conditions

We give the initial ratio (r) of the vertices with $+1$. In this paper, two types of initial conditions are applied for the initial state of the vertices. One is the random distribution, where the rN vertices out of the whole N vertices are randomly chosen and given $+1$. The other is selecting the vertices to be $+1$ corresponding to a local property of a network. Four centrality measures; degree centrality, closeness centrality, betweenness centrality [1,4] and clustering coefficient [5] are used in our numerical studies. The rN vertices with the largest centrality are given $+1$ as an initial state, while the rest $(1-r)N$ vertices are given -1 .

3 Numerical Studies Using Network Models

Numerical studies are performed using four theoretical, computer-generated network models; random graph, Barabási-Albert model [6], Klemm-Eguíluz model ($\mu = 0.1$) [7] and CNN (Connecting Nearest Neighbor) model [8], with the vertices $N = 36000$ and the average degree $\langle k \rangle = 10$. It is known that they all have 'small-world' property and the three network models except random graph have power-law degree distribution.

The numerical results are as follows. Typical results of the time-progress transition of the ratio of $+1$ are shown in Fig. 1. Fig. 2 shows the convergence ratio of the state $+1$ as a function of the initial ratio r , on each initial conditions. Some cases have an oscillation in convergence. We consider the average of the ratio of the last two step as a convergence ratio if it oscillates.

On random graph and BA model network, the $+1$ state has been extinguished or thoroughly spread in the whole network, while on KE model and CNN model two states are co-existing in convergence. On KE model with a certain initial conditions, a strong, damped oscillation mode is observed, while there is no or little oscillation on other networks. A dynamics and a convergence of the states are also affected by the centrality of the vertices which have initial states. It determines whether the state $+1$ will spread or deflate.

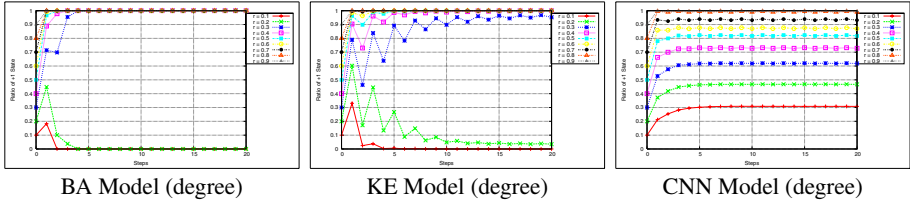


Fig. 1. Typical results of the transition of the ratio of the state +1 on network models. Each caption describes a network model and an initial condition.

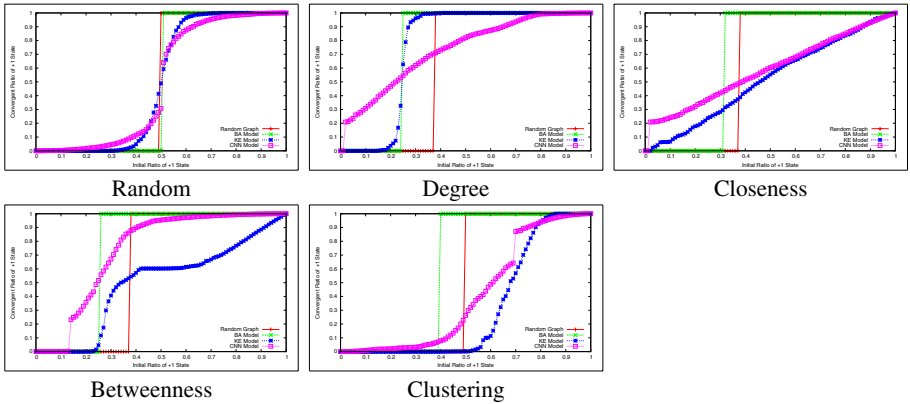


Fig. 2. Convergence ratio of the state +1 on network models by each initial conditions. Horizontal and vertical axes indicate the initial and the convergence ratio. Random graph, BA model, KE model and CNN model are represented as red, green, blue and purple dot, respectively.

4 Results

As shown in the previous section, the dynamics caused by our model strongly depends on the initial conditions and the network structures. It is found that patterns from the relations among initial condition, r and the convergence ratio are classified into several classes (Fig. 3).

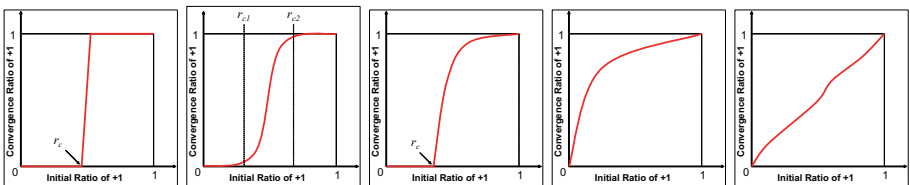


Fig. 3. Several classes are derived from the numerical results. Typical patterns are shown above.

On random graph and BA model, it is like a step function with all initial conditions, while convex curve or sigmoid-function-like curve are observed on KE model and CNN model with several initial conditions. In some cases, such as CNN model with betweenness-order initial distribution, they have non-trivial critical values to change the state. The absorbing state (all vertices take +1 or -1) in random graph and BA model might be related to low clustering coefficient. It is quite difficult to explain the oscillation or the co-existence of two states in fully developed stage from the conventional structural measures of network.

5 Conclusions

A new method to analyze a network structure from the patterns derived from the simple two-state diffusion process is proposed. Numerical studies are carried out using various network models. The phenomena appeared on the networks strongly depend on the initial conditions and the network structures. We classify the numerical results into several classes. The pattern in the classes may be connected with new structural properties.

It is considered that the diffusion dynamics by our method reveals some novel structural characteristics, and it gives us a clue for exploring the relations between a structure and a function of network.

References

1. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press (1994)
2. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* **45** (2003) 167 – 256
3. Tadić, B., Thurner, S.: Information super-diffusion on structured networks. *Physica A* (332) (2004) 566 – 584
4. Brandes, U.: A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* **25**(2) (2001) 163 – 177
5. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. *Nature* **393** (1998) 440 – 442
6. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286** (1999) 509 – 512
7. Klemm, K., Eguíluz, V.M.: Highly clustered scale-free networks. *Physical Review E* **65**(036123) (2002)
8. Vázquez, A.: Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E* **67**(056104) (2003)

Simulation of Micro-, Grand-, and Canonical Ensembles of Complex Networks

Christoly Biely and Stefan Thurner

Complex Systems Research Group, HNO, Medical University of Vienna,
Währinger Gürtel 18-20, A-1090 Vienna, Austria

and
Atominstytut der Österreichischen Universitäten,
Stadionallee 2, A-1020 Vienna, Austria

Abstract. The analysis of statistical ensembles of networks by means of simulation is an important possibility to explore networks which emerge by optimization of some 'fitness'-function. In this paper, we compare the situations of the micro-, grand- and canonical ensemble based on their respective partition functions. We present results for a specific, recently introduced Hamiltonian. Interestingly, for all three ensembles we find scale-free networks with 'complex' topology for a wide range of parameters. We further show results of some topological measures depending on energy and temperature.

1 Introduction

Properties of real-world networks and their exploration by analytical means and simulation finds increasing interest among physicists. Most of the real-world networks share typical structural key features such as non-random clustering, a scale-free (power-law) degree distribution and a small average path-length. A vast number of models have been proposed to catch these fundamental aspects [1, 2]. Predominantly, specific rewiring procedures and the evolution of networks associated therewith have been discussed, see e.g. [2, 3, 4], emphasizing the role of non-equilibrium processes involved. Orienting towards a statistical description of networks, the role of superstatistics [5] and network-Feynman graphs [6, 7] have been brought up. The statistical mechanics of equilibrium-networks has been discussed recently, theoretically [8, 9] as well as by simulation [10, 11]. The statistical-mechanics approach may provide significant insights into processes taking place in a network of approximately constant size or in systems, where the number of nodes can be supposed to change adiabatically, i.e. where the time-scale of rearrangements is much faster than the time-scale on which new nodes and links are added to the system. Recently, a number of interesting Hamiltonians have been introduced [3, 9, 10]. A particular form, inspired from utility considerations, has been reported to reproduce features of real-world-networks (including non-trivial scaling of the cluster-coefficient) [12]. While the latter article focused on the canonical ensemble, the present article tackles simulations of different ensembles, the micro- and grand- canonical in

particular. The paper is organized as follows: In section 2 we sum up the definition of relevant network-ensembles. In section 3 we present simulation algorithms in analogy to other lattice systems. Discussion of results obtained for a specific network-Hamiltonian is found in section 4. We conclude with a short discussion in section 5.

2 Statistical Mechanics of Networks

We consider simple, symmetric networks with a fixed number of distinguishable nodes $i = 1, \dots, N$, connected by a fixed number of $\ell = 1, \dots, L$ indistinguishable links. The network is represented by its adjacency matrix \mathbf{c} , where $c_{ij} = c_{ji} = 1$, if a link connects nodes i and j and $c_{ij} = c_{ji} = 0$, otherwise. Thus, we define the microcanonical partition function as (also see [8, 10, 12])

$$\mathcal{Z}_M(E, N, L) = \sum_{P(\mathbf{c})} \frac{1}{L!} \delta(E - \mathcal{H}(\mathbf{c})) \delta(L - \text{Tr}(\frac{\mathbf{c}^2}{2})) , \quad (1)$$

with $\mathcal{H}(\mathbf{c})$ being the network Hamiltonian and $P(\mathbf{c})$ denoting all permutations of the entries $c_{ij} = 1$ in $N \times N$ adjacency-matrix (not including permutations of the $c_{ij} = 0$ entries). The canonical partition may be derived from Eq. (1) via Laplace-Transformation, see e.g. [13], or via the maximum-entropy method, as proposed by Newman and Park in [9]:

$$\mathcal{Z}_C(\beta, N, L) = \sum_{P(\mathbf{c})} \frac{1}{L!} \delta\left(L - \text{Tr}(\frac{\mathbf{c}^2}{2})\right) e^{-\beta \mathcal{H}(\mathbf{c})} \quad (2)$$

The generalization to the grand-canonical partition function is straightforward. It can again be seen as Laplace transform or as maximum-entropy ensemble: It involves another thermodynamic potential (Lagrange-Parameter), namely the 'link-fugacity' μ :

$$\mathcal{Z}_G(\beta, N, \mu) = \sum_{P(\mathbf{c})} \frac{1}{L!} e^{L\mu\beta} e^{-\beta \mathcal{H}(\mathbf{c})} \quad (3)$$

Short discussions about the grand-canonical partition function can be found in [8, 10, 11].

3 Simulational Aspects

Irrespective of the ensemble definition applied, the 'core' of a computer-simulation of Eqs. (1), (2) or (3) will be the rewiring of a randomly chosen link between nodes i and j to a new link between randomly chosen nodes m and n , denoted in the following by $\ell(i, j) \rightarrow \ell(m, n)$. The choice of the random link to be replaced is best done with a 'master'-list comprising all the links currently existing in the system. Choosing a random link from a primarily randomly chosen node does not necessarily fulfill the condition of equal *a-priori* probabilities.

In the following, to exclude any methodological doubts, we shortly discuss the different procedures adopted to generate realizations of the three different ensembles investigated based on the rewiring $\ell(i, j) \rightarrow \ell(m, n)$. Although the discussion is similar to the one in [11], some important differences exist (in particular for the microcanonical ensemble which is defined as ensemble for a given degree distribution in [11]).

Microcanonical ensemble: Here, we have chosen the method proposed by Creutz [14], where a fictitious demon leads to extra one-dimensional randomness in the system. In close analogy to lattice systems, we have implemented the following scheme: (i) An initial random configuration with E_{rand} is chosen. The microcanonical energy E to be simulated is fixed and the demon energy is set $E_D = E - E_{rand}$. (ii) The energy change ΔE , associated with a random rewiring trial $\ell(i, j) \rightarrow \ell(m, n)$ is calculated. (iii) If $\Delta E < 0$, the rewiring is accepted and $E_D \rightarrow E_D + |\Delta U|$. If $\Delta E > 0$ it is checked whether $E_D > \Delta U$. If yes, the rewiring is accepted and the demon energy decreased: $E_D \rightarrow E_D - \Delta E$. Otherwise the rewiring is rejected.

Canonical ensemble: For the simulation of the canonical ensemble we have adopted the method introduced by Metropolis [15]. The application to networks has already been performed and discussed in previous works [7, 8, 9, 12]. The random rewiring $\ell(i, j) \rightarrow \ell(m, n)$ with associated energy-change ΔE is accepted with probability $\min\{1, \exp(-\beta(\mathcal{H}(\mathbf{c}) - \mathcal{H}(\hat{\mathbf{c}})))\}$, where $\mathcal{H}(\mathbf{c})$ is the Hamiltonian of the system, and $\hat{\mathbf{c}}$ is the adjacency matrix after a proposed rewiring step.

Grand-Canonical ensemble: Although some aspects of the grand-canonical have been discussed in [8, 10, 11], simulations of a specific hamiltonian have – to our knowledge – not been conducted so far. The necessary transition probabilities can be easily derived from the grand-canonical partition function (3) via the condition of detailed balance

$$\frac{P(\mathbf{c} \rightarrow \hat{\mathbf{c}})}{P(\hat{\mathbf{c}} \rightarrow \mathbf{c})} = \frac{\hat{L}! \binom{N(N-1)}{2} - \hat{L}!}{L! \binom{N(N-1)}{2} - L!} \exp\left(\beta(\mu\hat{L} - \mu L + \mathcal{H}(N, L, \mathbf{c}) - \mathcal{H}(N, \hat{L}, \hat{\mathbf{c}}))\right) \quad , \quad (4)$$

where \hat{L} is the new number of links in the system. Based on this expression, one can formulate the necessary Monte-Carlo step: A single update step consists of either adding a link (with probability of acceptance W_{+1}), removing a link (W_{-1}) or rewiring a link (W_0). The general expression for W_t follows from Eq. (4) and is given by

$$W_t = \min(1, s_t \exp(-\beta\Delta E)) \quad t = 0, +1, -1 \quad , \quad (5)$$

where

$$\begin{aligned} s_0 &= 1 \\ s_{+1} &= \frac{N(N-1)/2-L}{L+1} e^{-\beta\mu} \\ s_{-1} &= \frac{L}{N(N-1)/2-L+1} e^{\beta\mu} \quad . \end{aligned} \quad (6)$$

Calculation of energy-difference: It is apparent that in all three cases, calculation of $\hat{\mathcal{H}} - \mathcal{H}$ is of major importance and should therefore be carried out

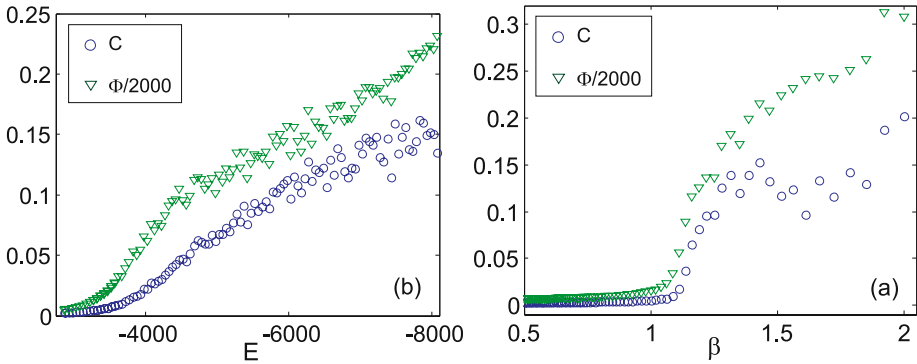


Fig. 1. Ensemble-average of cluster-coefficient $C(\mathbf{c})$ and maximal degree $\Phi(\mathbf{c})$ in dependence of energy for the micro-canonical ensemble (a) and in dependence of the inverse temperature β for the canonical ensemble (b). The data for Φ has been adjusted via division through 2×10^3 .

effectively. This means, that one would not like to calculate the energy difference via calculation of the total Hamiltonian of the adjacency matrix \mathbf{c} ,

$$\mathcal{H}(\mathbf{c}) = \sum_{\ell} \mathcal{H}(\ell) = \sum_i k_i \mathcal{H}(i) \quad , \quad (7)$$

where ℓ runs over all links and i over all nodes in the system, but only the energy change associated with a specific rewirement $\ell(i, j) \rightarrow \ell(m, n)$. Unfortunately, if the Hamiltonian depends e.g. on the degrees of both nodes, $\mathcal{H}(\ell) = \mathcal{H}(k_i, k_j)$, the correct expression of the energy-change is given by

$$\Delta E = \mathcal{H}(k_m + 1, k_n + 1) - \mathcal{H}(k_i, k_j) + \sum_{\tau \in \{k, i, m, n\}} \sum_{\xi \in N(\tau)} \Delta \mathcal{H}_{\tau\xi} \quad , \quad (8)$$

where

$$\Delta H_{\tau\xi} = \mathcal{H}(\hat{k}_{\tau}, k_{\xi}) - \mathcal{H}(k_{\tau}, k_{\xi}) \quad , \quad (9)$$

and \hat{k}_{τ} is the new degree of node τ . Note, that – of course – additional specialization is necessary if the randomly chosen nodes m and n and/or the randomly chosen nodes i and k are neighbours of each other. In these cases, Eq. (8) has to be modified to the different ‘special’ cases. Practically, (8) implies that as soon as $N(i) + N(j) + N(m) + N(n) \approx L$ holds for any specific rewirement $\ell(i, j) \rightarrow \ell(m, n)$, the effort to correctly calculate the energy-change amounts to the re-calculation of the total energy sum as given by Eq. (7). Complementary, for $N(i) + N(j) + N(m) + N(n) < L$ the energy may be calculated by implementation of Eq. (8) more efficiently.

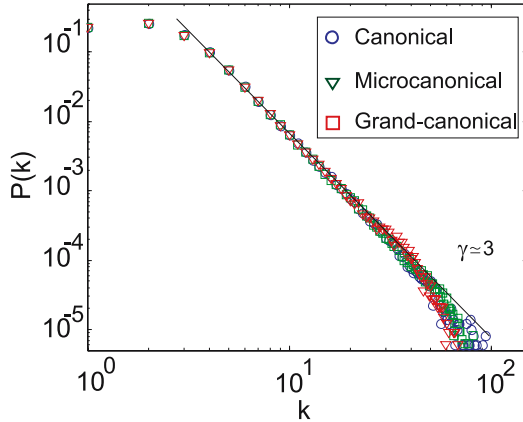


Fig. 2. Ensemble equivalence of the degree-distribution for the micro-canonical (triangles), canonical (squares) and grand-canonical (circles) ensemble for $N = 1000$, $L = 3000$ and appropriately chosen parameters (see text)

4 Results for a Specific Hamiltonian

For specific simulations of the ensembles described above we have chosen the Hamiltonian introduced in [12]:

$$\mathcal{H} = c - \sum_{\ell} \log(b + \Delta k_{\ell}) \quad , \quad (10)$$

where $\Delta k_{\ell} = |k_i - k_j|$ (i and j being the nodes link ℓ is joining). b and c are shape-parameters. While the form is somehow similar to a Hamiltonian studied in [10], Eq. (10) explicitly introduces a form where the energy of the system can not be expressed as a sum over nodes with the respective contributions only being dependent on the own degree. We speculate, that it is exactly this circumstance, which also produces scaling of the cluster-coefficient $\langle C(k) \rangle \sim k^{-1}$ [16]. We simulated micro-, and canonical partition functions of (10) (canonical partition functions have already been simulated in [12]) with $c = 0$, $b = 5$ and grand-canonical partition functions with $c = 10$ and $b = 5$. Typically, we have simulated systems of $N = 10^3$ nodes for about $T = 2 \times 10^6$ time-steps, where we averaged over every $20 \times N$ th configuration of the network. To eliminate dangers of a too high sampling frequency we have also compared individual runs to simulations amounting to 9×10^7 rewirings with averages done over every 10^6 th rewiring step ($\approx 700L$). No significant discrepancies to the results shown have been found. It has already been stated, that this Hamiltonian exhibits a behaviour of the internal energy $U(T)$ indicating the existence of a critical point. Therefore we do not discuss these aspects here. Instead, we show the maximum degree $\Phi(c)$ of a specific network ensemble and the average cluster coefficient C in dependence of energy (temperature) for the micro-canonical (canonical) ensemble

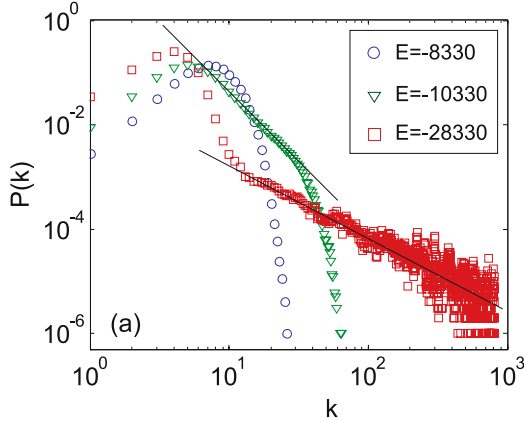


Fig. 3. (a) Some chosen microcanonical ensemble averages of degree distributions for $N = 1000$ and $L = 8000$ at different energies

in Figure 1. $\Phi(c)$ has already been proposed as order parameter when topological transitions are involved in preceding work [10]. For the micro-canonical ensemble, the two values first grow approximately exponential. At $E \approx -4500$, the behaviour changes as the network is no longer dominated by a homogenous mass of nodes, but by 'star-like' nodes and comparatively low-connected nodes (still, $\langle C(k) \rangle \sim k^{-1}$ is preserved for all ensembles). In Figure 2 we demonstrate the ensemble equivalence for the microcanonical, the grand-canonical and the canonical ensemble. The fixed energy of the microcanonical ensemble was chosen to be equal to the expectation value of the energy of the canonical ensemble corresponding to the scale-free regime of the degree-distribution (at $T \approx 0.94$ and $\langle E \rangle \approx -3530$ units). At this point, we mention that we found the ground-state of microcanonical ensemble at about $E \approx -9750$ units, where the demon's energy does no longer fully 'flow' into the network. For the grand-canonical degree-distribution we simulated at $T \approx 0.94$ and $\mu \approx 0.2$, recovering the energy average $E \approx -3570$ units (after subtracting the contribution resulting from $c \neq 0$) and the link average $\langle L \rangle \approx 3020$. Additionally, we adapted the parameter c to a value where the number of links in the system is stable ($c = 8.0$). At $c = 0$, the energy-decrease realized by adding a link is too large to allow stabilization of the dynamics. Setting $c \neq 0$ implies, that every link has a constant cost.

As expected, the ensembles show a nice equivalence. Naturally, also the behaviour for $\langle C(k) \rangle$ is very similar, exhibiting scaling of $\langle C(k) \rangle \sim k^{-1}$.

Further, we present another interesting microcanonical ensemble average of networks obtained by the Hamiltonian (10). In [12], it has been pointed out, that a characteristic scale emerges for increasing number of links L in the system. Figure 3 shows microcanonical ensemble averages for $L = 8000$, $N = 1000$ at various energies. From the degree distribution we see that the network is organized in a scale free manner over two orders of magnitude at $E \approx -28 \times 10^3$ and over one order of magnitude at $E \approx -10 \times 10^3$. Degrees less than 10 are

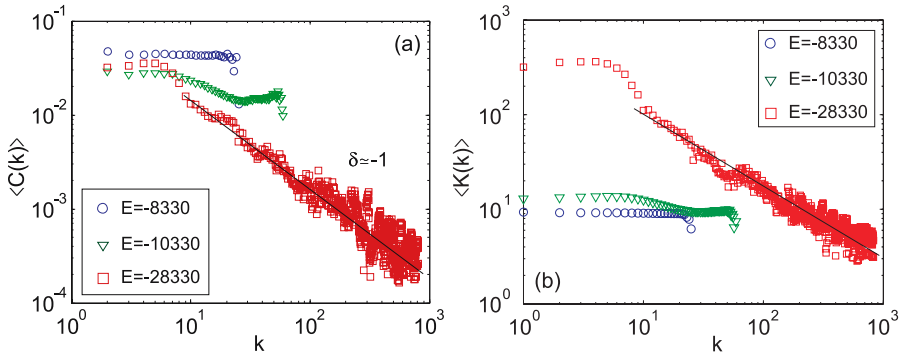


Fig. 4. (a) Some chosen microcanonical ensemble averages of the degree-dependence of the Cluster-coefficient distributions $\langle C \rangle$ for $N = 1000$ and $L = 8000$ at different energies. (b) Respective ensemble averages for the neighbour-connectivities $\langle K(k) \rangle$.

dominated by Poissonian behaviour for $E \approx -28 \times 10^3$. In Figure 4, we show ensemble averages of $\langle C(k) \rangle$ and of the average neighbour connectivity $\langle K(k) \rangle$ at the same energies. These curves show the existence of a highly non-trivial network-structure, perfectly reproducing the empirical findings for many socio-economic networks where the scaling $\langle C(k) \rangle \sim k^{-\delta}$ with $\delta \approx 1$ is observed [16]. It also becomes evident, that the characteristic scaling of $\langle C(k) \rangle \sim k^{-1}$ does not depend on the scaling of the degree distribution, thus exhibiting a type of universal behaviour for its own, see [16] for a brief discussion.

5 Conclusion

We have presented simulations of microcanonical, canonical and grand-canonical ensembles of networks based on strict application of standard classical statistical mechanics to networks. Based on a previously introduced Hamiltonian [12], we have documented that the emergence of typical 'complex' networks is not bound to the canonical ensemble, as expected. We have also presented results of changes of topological 'indicators' along the energies/temperatures involved in the system. We have further shown that complex organized networks may be obtained in other regimes of the system than previously explored (for higher L and lower energy). In summary, we have provided clear evidence that the architecture of real-world networks exhibits – to some extent – patterns not untypical for equilibrium.

References

1. Barabasi, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74** (2002) 47.
2. Dorogovtsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press (2003).

3. Baiesi, M., Manna, S.S.: Scale-Free Networks from a Hamiltonian Dynamics. *Phys. Rev. E* **68** (2003) 047103.
4. Alava, M.J., Dorogovtsev, S.N.: Complex networks created by aggregation. *Phys. Rev. E* **71** (2005) 036107.
5. Abe, S., Thurner, S.: Complex networks emerging from fluctuating random graphs: Analytic formula for the hidden variable distribution. *Phys. Rev. E* **72** (2005) 036102.
6. Dorogovtsev, S.N., Mendes, J.F.F., Samukhin A.N.: Principles of statistical mechanics of uncorrelated random networks. *Nucl. Phys. B* **666** (2003) 396.
7. Burda, Z., Correia, J.D., Krzywicki, A.: Statistical ensemble of scale-free random graphs. *Phys. Rev. E* **64** (2001) 046118.
8. Berg, J., Lässig, M.: Correlated random networks. *Phys. Rev. Lett.* **89** (2002) 228701.
9. Park, J., Newman, M.E.J.: The Statistical Mechanics of Networks. *Phys. Rev. E* **70** (2004) 066117.
10. Farkas, I., Derenyi, I., Palla, G., Vicsek, T.: Equilibrium statistical mechanics of network structures. *Lect. Notes Phys.* **650** (2004) 163; Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Statistical mechanics of topological phase transitions in networks. *Phys. Rev. E* **69** (2004) 046117.
11. Bogacz, L., Burda, Z., Waclaw, B.: Homogeneous complex networks. *cond-mat/0502124*.
12. Biely, C., Thurner, S.: Statistical Mechanics of complex networks at a critical point: Complexity, without Irreversibility? Santa Fe Working Paper 05-10-038.
13. Grandy, W.T.: *Foundations of Statistical Mechanics*. Kluwer Academic Publishers, 1987.
14. Creutz, M.: Microcanonical Monte Carlo Simulation. *Phys. Rev. Lett.* **50** (1993) 1411-1414.
15. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, H., Teller, E.: Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6) (1953) 1087.
16. Ravasz, E. and Barabasi, A.L.: Hierarchical Organization of Complex Networks. *Phys. Rev. E* **67** (2003) 026112.

Synchronization in Network Structures: Entangled Topology as Optimal Architecture for Network Design

Luca Donetti^{1,3}, Pablo I. Hurtado^{1,2}, and Miguel A. Muñoz¹

¹ Departamento de Electromagnetismo y Física de la Materia, and Instituto *Carlos I*
de Física Teórica y Computacional

Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain

² Laboratoire des Colloïdes, Verres et Nanomatériaux,

Université Montpellier II, Montpellier 34095, CEDEX 5 France

³ Departamento de Electrónica y Tecnología de Computadores

Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain

donetti@ugr.es, phurtado@onsager.ugr.es, mamunoz@onsager.ugr.es

Abstract. In these notes we study synchronizability of dynamical processes defined on complex networks as well as its interplay with network topology. Building from a recent work by Barahona and Pecora [Phys. Rev. Lett. **89**, 054101 (2002)], we use a simulated annealing algorithm to construct optimally-synchronizable networks. The resulting structures, known as *entangled networks*, are characterized by an extremely homogeneous and interwoven topology: degree, distance, and betweenness distributions are all very narrow, with short average distances, large loops, and small modularity. Entangled networks exhibit an excellent (almost optimal) performance with respect to other flow or connectivity properties such as robustness, random walk minimal first-passage times, and good searchability. All this converts entangled networks in a powerful concept with optimal properties in many respects.

1 Introduction

It is broadly recognized that most complex systems in Nature are organized as intricate network patterns [1, 2]. This observation has triggered an intense research effort aimed at understanding the organizing principles of these networks, their structural properties, and the interplay between topology and dynamics [1, 2]. It was recently recognized that the classical models of random networks developed in graph theory were unable to describe the random but structured, hierarchical network patterns found in Nature. Since then, a number of paradigmatic models (as small-world and scale-free nets [2]) have seen the light. They mimic some of the striking properties observed in real complex networks. In any case, network structures play an important role in many contexts ranging from brain neural circuits, cellular function webs, ecosystems, social networks, food webs, etc., to power grids, Internet or the world wide web. While most of

the initial effort was put into understanding the topological properties of networks, the interest has gradually shifted towards the analysis of the interplay between topology and the dynamics of network components. In general, each element (node) in a network undergoes a dynamical process while coupled to other nodes. The system collective behavior depends strongly on the efficiency of communication paths, which is in turn dictated by the underlying network topology. In this way, the network structure determines to a large extent the possibility of a coherent response.

Complete synchronization is the most prominent example of coherent behavior, and is a key phenomenon in systems of coupled oscillators as those characterizing most biological networks or physiological functions [3]. For instance, synchronized neural firing has been suggested as specially relevant for neural signal transmission [4]. From a more technological point of view, precision synchronization of computer clocks in local area networks and the Internet is essential for optimal network performance. Moreover, in an interesting twist, the dynamics toward synchronization has been recently used as a dynamical process unveiling the underlying community structure in complex networks [5].

Here we study how synchronous behavior is affected by the network structure. The range of stability of a synchronized state is a measure of the system ability to yield a coherent response and to distribute information efficiently among its elements, while a loss of stability fosters pattern formation [6]. Here we answer the following question: *which is the topology that maximizes the network synchronizability?* [7]. We will construct such optimal topologies, for any fixed number of nodes and links, by employing an optimization procedure. The resulting structures, that we call *entangled networks*, are optimal not only for synchronizability, but also regarding other flow or connectivity properties.

The paper is structured as follows. In section 2 we summarize the spectral approach to synchronization, following Ref. [6]. In section 3 we introduce the optimization procedure to obtain networks with optimal synchronizability. Section 4 discusses the relation between the emerging structures and other optimal network designs in the literature. Finally, conclusions and further developments are presented. A shorter presentation of this work has been published before [7].

2 Spectral Approach to Synchronization in Networks

Consider N identical oscillators at the nodes of an undirected and unweighted graph. The state of an oscillator is represented in general by a vector \mathbf{x}_i , $i \in [1, N]$, where N is the number of nodes. The network is characterized by its Laplacian matrix \mathbf{L} , with elements $L_{ii} = k_i$ (the degree of node i), $L_{ij} = -1$ if nodes i and j are connected, and $L_{ij} = 0$ otherwise. \mathbf{L} is therefore a symmetric matrix with zero-sum rows and real, non-negative spectrum. The dynamics of the i -th node can then be represented in a very general form as,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{F}(\mathbf{x}_i) - \sigma \sum_{j=1}^N L_{ij} \mathbf{H}(\mathbf{x}_j) \quad . \quad (1)$$

Here $\mathbf{F}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ are unspecified evolution and coupling functions, respectively. In particular, $\mathbf{F}(\mathbf{x})$ controls the dynamics of the uncoupled oscillators, while $\mathbf{H}(\mathbf{x})$ specifies how variables at different nodes couple together. Most dynamical processes studied in the literature regarding synchronization can be recasted in forms equivalent to eq. (1) (see [6] for more general couplings).

In the synchronized state all oscillators behave identically at all times. That is, $\mathbf{x}_i(t) = \mathbf{x}^s(t) \forall i \in [1, N]$, where $\mathbf{x}^s(t)$ is solution of the uncoupled equation $\dot{\mathbf{x}}^s = \mathbf{F}(\mathbf{x}^s)$ ($\dot{\mathbf{x}}$ represents the time derivative of \mathbf{x}). The $N - 1$ synchronization constraints $\mathbf{x}_1(t) = \mathbf{x}_2(t) = \dots = \mathbf{x}_N(t)$ define a *synchronization manifold*. This manifold is invariant owing to the zero-sum row condition in the Laplacian matrix \mathbf{L} [6]. We are interested here in the stability of the synchronized state. For that, we introduce small perturbations ξ_i such that $\mathbf{x}_i = \mathbf{x}^s + \xi_i$, and expand to first order to arrive at: $\dot{\xi}_i = \sum_{j=1}^N [\partial \mathbf{F}(\mathbf{x}^s) \delta_{ij} - \sigma L_{ij} \partial \mathbf{H}(\mathbf{x}^s)] \cdot \xi_j$, where $\partial \mathbf{M}$ stands for the Jacobian of a matrix \mathbf{M} . Diagonalization of \mathbf{L} transforms these equations into a set of N independent equations for the normal modes [6, 7]:

$$\frac{d\mathbf{y}_k}{dt} = [\partial \mathbf{F}(\mathbf{x}^s) - \sigma \lambda_k \partial \mathbf{H}(\mathbf{x}^s)] \cdot \mathbf{y}_k \quad , \quad (2)$$

where $\lambda_k, k \in [1, N]$, are the eigenvalues of \mathbf{L} , $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. All the resulting equations have the same form $\dot{\mathbf{y}} = [\partial \mathbf{F}(\mathbf{x}^s) - \alpha \partial \mathbf{H}(\mathbf{x}^s)] \cdot \mathbf{y}$ for some positive constant α . The synchronized state \mathbf{x}^s will be stable if and only if all the perturbations fade away in time. This is equivalent to demanding the maximum Lyapunov exponent $\eta_{max}(\alpha)$ associated with the normal modes orthogonal to the synchronization manifold to be negative. The function $\eta_{max}(\alpha)$ has been called the *master stability function* in literature [6], and its dependence on α has an universal “V-shape” for most oscillating systems. In particular, $\eta_{max}(\alpha)$ is negative only in an interval $\alpha_A < \alpha < \alpha_B$. The synchronized state will be stable if all the non-trivial eigenvalues of \mathbf{L} , $\{\lambda_k : k = 2, \dots, N\}$, lie within the interval $[\alpha_A/\sigma, \alpha_B/\sigma]$. The following inequality then guarantees that there always exists a coupling strength σ for which the synchronized state is stable,

$$Q \equiv \frac{\lambda_N}{\lambda_2} < \frac{\alpha_B}{\alpha_A} \quad . \quad (3)$$

It is important to notice that the left hand side in the above inequality depends exclusively on the network topology, while the right hand side depends only on the dynamics (through \mathbf{x}^s , \mathbf{F} and \mathbf{H}). The σ range for which the synchronized state is stable, $\sigma \in [\frac{\alpha_A}{\lambda_2}, \frac{\alpha_B}{Q\lambda_2}]$, is larger for smaller eigenratios Q . In this way, networks with very small Q will exhibit very good (robust) synchronization properties for a generic dynamics. The aim of this paper is to find and characterize network topologies minimizing the eigenratio Q .

3 Optimizing Synchronizability: Entangled Networks

Most studies up to now have explored the value of the eigenratio Q for different pre-existing network topologies found in literature, as for instance small-world

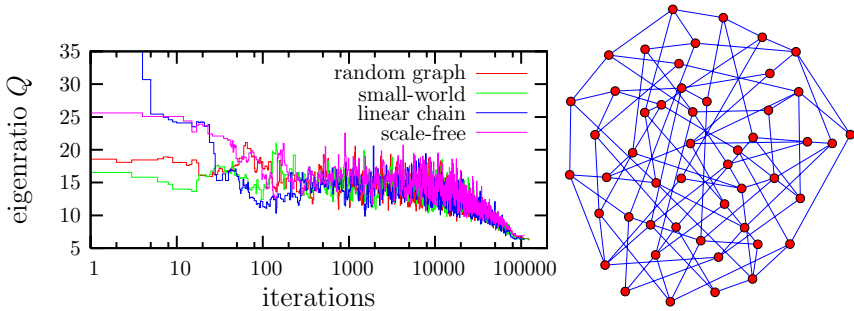


Fig. 1. Evolution of the eigenratio Q during the optimization procedure for different initial conditions. Here $N = 50$, $\langle k \rangle = 4$. In all cases, the algorithm converges to very homogeneous networks as the one depicted, with very similar values of Q .

or scale-free networks, trying to identify key topological features affecting Q . In this way, it has been reported that small-worlds have smaller Q than regular or purely random graphs [6], and this has been attributed to a smaller average distance between nodes in small-worlds. However, other works [8] have concluded recently that Q decreases as some heterogeneity measures decrease, even if the average distance between nodes increases in the process. On the other hand, synchronizability is enhanced in weighted complex networks [9].

In this paper we undertake a constructive approach to determine the network topology that optimizes synchronization. In order to do so, we devise a modified simulated annealing algorithm [10] to numerically minimize Q . We start from graphs with N nodes and a fixed average degree $\langle k \rangle$. At each step, a new graph is obtained by the random deletion of m links and addition of m new ones, where m is extracted from an exponentially decaying distribution [7]. The new graph is rejected if the resulting network is disconnected; otherwise, it is accepted with probability $p = \min(1, [1 - (1 - q)\delta Q/T]^{1/(1-q)})$, where $\delta Q = Q_{final} - Q_{initial}$ is the eigenratio change in the rewiring, and T is a temperature-like parameter. For $q \rightarrow 1$ we recover the standard Metropolis algorithm with *Hamiltonian* Q , while $q = -3$ turns out to be the most efficient value (results do not depend on the choice of the deformation parameter q , but convergence times do [10]). The first N rewirings are performed at $T = \infty$, and they are used to calculate a new T such that the largest δQ among the first N ones would be accepted with a large probability. T is kept fixed for $100N$ rewiring attempts or $10N$ accepted ones, whichever occurs first. Then T is decreased by 10% and the process is repeated until there are no more changes during five successive temperature steps, assuming in this case that the optimal network topology has been found. Most of these details can be modified without affecting the final outcome. The major drawback in the algorithm is that Q is a global observable slow to compute. For small enough N (≤ 30), the emerging optimal topology found is unique, while for larger N (we have optimized networks with N up to 2000) the output may change slightly from run to run, meaning that the eigenratio absolute minimum is not always reached due to the presence of metastable states. Nevertheless, the

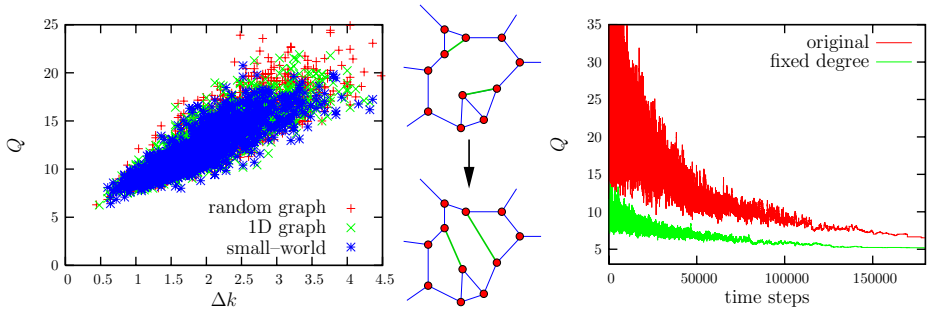


Fig. 2. Left: Q vs standard deviation of the degree distribution for $N = 50$, $\langle k \rangle = 4$ and three different initial conditions. Center: sketch of pair rewiring trial. Right: Eigenratio Q vs algorithmic steps for both minimization procedures (see text).

final values of Q are very similar for different runs (see Fig. 1), meaning that a reasonably good approximation to the optimal topology is always found [7].

We measure different topological observables during the minimization process to unveil the main traits of the emerging structures. In simple terms, we observe that as Q decreases the network becomes more and more homogeneous. This means that the standard deviation of distributions of most topological observables decreases as Q decreases. This is true in particular for the node degree distribution, see Fig. 2.a. We have used this degree homogeneity to improve the efficiency our optimization procedure by initializing the algorithm with regular networks (i.e. all nodes with the same degree), and restricting the rewiring steps to changes that leave the degree of each node unchanged (by randomly selecting *pairs* of links and exchanging their endpoints; see Fig. 2.b). The resulting algorithm converges much faster to the optimal network, and yields lower final eigenratios Q when the original one get trapped in a metastable state (Fig 2.c).

In Figs. 3.a-b we show the standard deviation of the average node-to-node distance and average betweenness, respectively, versus Q during an optimization run started from a random regular graph. Both observables exhibit the aforementioned tendency towards homogeneity. Particularly remarkable is the narrow betweenness distribution (Fig. 3.b), which is in marked contrast with the broad betweenness distributions observed in networks with strong community structure [11]. In addition, the averaged distance and betweenness also tend to decrease with Q , though they are less sensitive than their corresponding standard deviations, see Figs. 3.c-d. Another key feature of the optimal structures is the absence of short loops. This can be characterized by the *girth* (length of the shortest loop) or, better, via the average size of the shortest loop passing through each node. This last magnitude is shown in Fig. 3.e, where it is evident that the optimal network has very large average shortest loops. In particular, the clustering coefficient is zero for the optimal nets since no triangles are present.

In general, we call the emerging optimal structures *entangled networks*: all sites are very much alike (strong homogeneity) and the links form a very

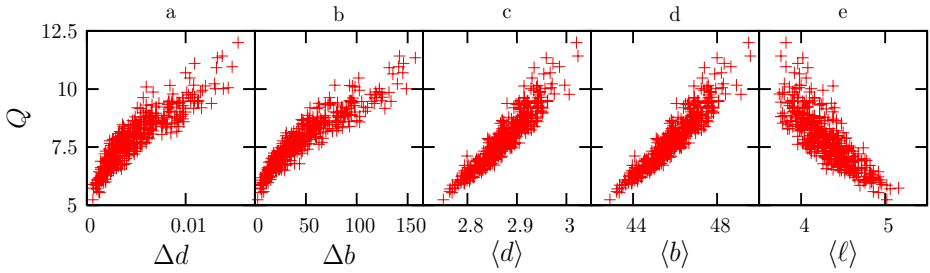


Fig. 3. Standard deviations of the node distance distribution (a), standard deviation of the betweenness distribution (b), average distance (c), average betweenness (d), and shortest-loop average length (e) as a function of the eigenratio Q

intricated or interwoven structure (lack of communities, poor modularity, and large shortest loops). Every single site is close to any other one (short average distances) owing not to the existence of intermediate highly connected hubs (as in scale free networks), but as a result of a very “democratic” or entangled structure in which properties such as site-to-site distance, betweenness, and minimum-loop size are very homogeneously distributed (see Figs. 1.b, 3.a-b). Sharp distributions are also typical of random graph, where randomness alone produces a statistical homogeneity. However in this case, a much stronger homogeneity is produced during the optimization of Q , as Figures 3.a-b show.

4 Relation to Other Optimal Topologies

A natural question concerns the relation between entangled networks and other optimal architectures found in the literature. For instance, recent work [12] has focused on the optimization of network robustness against random and/or intentional removal of nodes (attacks). For random graphs in the large- N limit, it is concluded that the most robust networks are obtained when the degree distribution only has a few peaks. In particular, random k -regular graphs turn out to be the global optimal solution against both errors and attacks in the robustness-optimization problem [12]. In this case, the error (f_r) and attack (f_a) percolation thresholds coincide, $f_r = f_a \equiv f_c(N, k)$, with $f_c(N, k) < f_c(\infty, k) = (k - 2)/(k - 1)$. A natural question now is whether further Q -minimization of these random regular graphs has some effect on the network robustness. As shown in [7] the minimization of Q improves significantly the network robustness, confirming that entangled networks are optimal from the robustness point of view. This is because entangled topologies include correlations, absent in random networks, which enhance their resilience. In addition, there is also evidence that networks with properties similar to those of entangled graphs maximize reliability against link removal [7].

Different models of traffic flows on complex network have been recently studied [13, 14]. In principle highly inhomogeneous scale-free networks perform well when the traffic is low; hubs can provide fast transition times, while they easily

jam when the traffic increases. With the model of [14] it has been shown that if the density of traveling information packets is above a given threshold, the optimal network topology is a highly homogeneous, isotropic configuration, strongly resembling entangled graphs. In a similar way, it has been recently reported [15] that the interplay between network growth processes and evolutionary selection rules gives rise in some cases to very homogeneous structures with large minimal-loops that strongly resemble entangled networks (see Fig. 3.c in [15]).

Also, during our optimization procedure, λ_N is observed to change very little with respect to λ_2 , and therefore, minimizing Q is equivalent for all practical purposes to maximizing λ_2 . This provides another interesting connection with graph theory, where it is known that regular graphs with a large λ_2 (i.e. *large spectral gap*), are good *expanders* (see [16, 7] for a definition and applications). Expander graphs are very important in computer science for various applications (as the design of efficient communication networks or construction of error-correcting codes) and can be proved to exhibit a rapid decay of random-walk distributions towards their stationary state [17]. This converts entangled graphs in (almost) optimal for many information flow processes.

5 Summary and Outlook

We have introduced the concept of “entangled networks” [7]. These are constructed using an optimization principle by imposing the eigenvalues of the Laplacian matrix to have a quotient λ_N/λ_2 as small as possible, guaranteeing in this way a robust synchronizability and coherent behavior. The emerging topologies are extremely homogeneous: all nodes look very much alike (constituting a topology radically distinct from scale free networks). Also, the node-to-node average distance tends to be small while the average shortest loops are very large, and there is no modular (or community) structure. Entangled networks exhibit optimal synchronization properties, but they are also optimal or almost-optimal for other communication or flow properties: robustness and resilience against errors and attacks, traffic flow in the presence of congestion, relaxation properties of random walks, etc. These connections make of entangled networks a key tool in the context of complex networks.

An interesting issue concerns the existence of entangled networks in Nature. Their construction requires a global optimization process which is unlikely to occur in natural evolving systems. Presently, we are working on the identification of local evolutionary rules which give rise to locally-optimal synchronizable network patterns, or other feasible approximations to entangled networks.

Acknowledgments

We thank D. Cassi and P.L. Krapivsky for useful discussions, and B. Tadić and S. Thurner for inviting us to the ICCS06. Financial support from the Spanish MCyT under project No. FIS2005-00791, EU COSIN-project-IST2001-33555, and EU HPRN-CT-2002-00307 (DYGLAGEMEM) are also acknowledged.

References

1. Barabási, A.-L.: *Linked: The New Science of Networks*, Perseus Publishing, Cambridge (2002); Pastor-Satorras, R., Vespignani, A.: *Evolution and Structure of the Internet*, Cambridge Univ. Press, Cambridge (2004).
2. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of Networks, *Adv. Phys.* **51**, 1079-1187 (2002); Albert, R., Barabási, A.-L.: Statistical Mechanics of Complex Networks, *Rev. Mod. Phys.* **74**, 47-97 (2002). Krapivsky, P.L., Redner, S.: A Statistical Physics Perspective on Web Growth, *Computer Networks* **39**, 261-276 (2002).
3. Pikovsky, A., Rosenblum, M., Kurths, J.: *Synchronization: A Universal Concept in Nonlinear Sciences*, Cambridge University Press, Cambridge (2001); Glass, L.: Synchronization and Rhythmic Processes in Physiology, *Nature* **410**, 277-284 (2001).
4. Belykh, I., de Lange, E., Hasler, M.: Synchronization of Bursting Neurons: What Matters in the Network Topology, *Phys. Rev. Lett.* **94**, 188101 (2005); Cossart, R., Aronov, D., Yuste, R.: Attractor Dynamics of Network Up States in the Neocortex, *Nature* **423**, 283-288 (2003).
5. Arenas, A., Díaz-Guilera, A., Pérez-Vicente, C.J.: Synchronization Reveals Topological Scales in Complex Networks, *ArXiv cond-mat/0511730*.
6. Barahona, M., Pecora, L.M.: Synchronization in Small-World Systems, *Phys. Rev. Lett.* **89**, 054101 (2002); Pecora, L.M., Carrol, T.L.: Master Stability Functions for Synchronized Coupled Systems, *Phys. Rev. Lett.* **80**, 2109-2112 (1998); Pecora, L.M., Barahona, M.: Synchronization of Oscillators in Complex Networks, *Chaos and Complexity Letters*, **1**(1), 61-91 (2005).
7. Donetti, L., Hurtado, P.I., Muñoz, M.A.: Entangled Networks, Synchronization, and Optimal Network Topology, *Phys. Rev. Lett.* **95**, 188701 (2005).
8. Nishikawa, T., et al. : Heterogeneity in Oscillator Networks: Are Smaller Worlds Easier to Synchronize?, *Phys. Rev. Lett.* **91**, 014101 (2003); Hong, H., et al.: Factors that Predict Better Synchronizability on Complex Networks, *Phys. Rev. E* **69**, 067105 (2004).
9. Chavez, M. et al. : Synchronization is Enhanced in Weighted Complex Networks, *Phys. Rev. Lett.* **94**, 218701 (2005).
10. Penna, T.J.P.: Traveling Salesman Problem and Tsallis Statistics, *Phys. Rev. E* **51**, R1-R3 (1995).
11. L. Donetti and M. A. Muñoz, Detecting Network Communities: a new systematic and powerful algorithm, *J. Stat. Mech.: Theor. Exp.* (2004) P10012-P10027.
12. Valente, A.X.C.N., Sarkar, A., Stone, H.A.: Two-Peak and Three-Peak Optimal Complex Networks, *Phys. Rev. Lett.* **92**, 118702 (2004).
13. B. Tadić, S. Thurner, G. J. Rodgers: Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations, *Phys. Rev. E*, **69**, 036102 (2004).
14. Guimerà, R., et al. : Optimal Network Topologies for Local Search with Congestion, *Phys. Rev. Lett.* **89**, 248701 (2002).
15. Colizza, V. et al.: Network Structures from Selection Principles, *Phys. Rev. Lett.* **92**, 198701 (2004).
16. Sarnak, P.: What is an Expander?, *Not. Am. Math. Soc.* **51**, 762-763 (2004).
17. Lovasz, L.: Random Walks on Graphs: A Survey, in *Combinatorics, Paul Erdős is Eighty*, vol. 2, pp. 1-46, Keszthely, Hungary (1993).

Dynamics of Content-Based Networks

Duygu Balcan¹ and Ayşe Erzan^{1,2}

¹ Department of Physics, Faculty of Sciences and Letters
Istanbul Technical University,
Maslak 34469, Istanbul, Turkey

`balcand@itu.edu.tr`, `erzan@itu.edu.tr`

² Gürsey Institute, P.O.B. 6, Çengelköy 34680, Istanbul, Turkey
`erzan@gursey.gov.tr`

Abstract. Content-based networks are introduced and their topological properties are outlined. A content-based model with Random Boolean dynamics, designed to mimic the gene regulation network, exhibits an increase in the number and complexity of attractors for increasing number of nodes. However, contrary to expectations based on Mean Field calculations for random scale-free networks, the attractors are not chaotic, even for average connectivities in excess of 2. Thus, the present model offers a promising tool for understanding complex biological networks.

1 Introduction

Complex networks [1, 2, 3, 4] have become an essential tool for understanding coarse grained properties of discrete, distributed systems with often highly specific interactions as, for example, in genomic or proteomic networks [5, 6], which rely on recognition and binding mechanisms. Independently of the nature of the interaction, we submit that the specificity of the network connections can be modelled by requiring a certain amount of information to be shared between the nodes. We have coined the term “content-based” to denote networks where a certain information content coded in a common alphabet characterises each node. In two previous papers [7, 8] a null-version of such a model with random strings of letters, was formulated and simulations as well as analytical results regarding its topological properties were provided. (An earlier, more ambitious version of such a model was proposed by Reil [9].) In the present paper, we extend our model to incorporate certain features of the gene regulation network in a more realistic fashion [10, 11], and also include a dynamics, provided by randomly assigned Boolean functions at the nodes, in the same spirit as the Random Boolean Network (RBN) models introduced by Kauffman [12, 13].

In the following section we outline our content-based model and summarise earlier results. In Section 3 the details of the extended model and the dynamics will be introduced. In Section 4, preliminary results on the dynamics, and in Section 5 a short discussion will be presented.

2 A Content-Based Null Model

In our original model a linear code of length L , which can be thought of as an “artificial chromosome,” has elements which are randomly assigned from an alphabet containing $r + 1$ letters with the probability distribution $P(x)$,

$$P(x) = p\delta(x - r) + \frac{1 - p}{r} \sum_{\rho=0}^{r-1} \delta(x - \rho) \quad . \quad (1)$$

Each “gene”, G_i , is defined as the string between two successive occurrences of the letter “ r ” on the chromosome, $G_i \equiv (x_{i,1}, \dots, x_{i,j}, \dots, x_{i,l_i})$ where $x_{i,j} \neq r$ and l_i is the length of the i th gene. This results in an exponential distribution of string lengths, $P(l) \propto (1 - p)^l$. The model chromosome thus has only three parameters, the length L , besides p and r appearing in the above distribution.

We now associate a network with this model chromosome, by assigning a node to each “gene,” and postulating an interaction matrix w_{ij} based on the string matching condition, thus:

$$w_{ij} = \begin{cases} 1 & \text{if } G_i \subset G_j \\ 0 & \text{otherwise} \end{cases} \quad . \quad (2)$$

It should be noted that the linear codes associated with the “genes” do not necessarily represent a nucleotide sequence but rather should be considered as the information content of that node. The probability of finding an edge between two nodes with string-lengths l and k , directed from the first to the second, is $p(l, k) = 1 - (1 - r^{-l})^{k-l+1}$.

The resulting network is small-world type, having a large clustering coefficient and very small average shortest path length.[7] It is very robust to random removal of nodes, and we conjecture that the percolation threshold is 0, as for scale-free networks with exponents $\gamma \leq 3$ [3, 14]. Moreover, the network exhibits “Highly Optimised Tolerance” as defined by Carlson and Doyle [15].

The in- and out-degree distributions are displayed in Fig.1. The out-degree distribution has a continuous regime, decaying as a power law with an exponent γ_1 , followed by an oscillatory part whose envelope with an exponent γ_2 where [8]

$$\gamma_2 = \frac{1 \ln r + \ln(1 - p)}{2 \ln r - \ln(1 - p)} \quad , \quad \text{and} \quad \gamma_1 = \frac{1}{2} + \gamma_2 \quad . \quad (3)$$

The length distribution of the sequences identified with the nodes changes the topology of the network. Taking a Gaussian distribution for the lengths yields in- and out-degree distributions both similar to that shown in Fig. 1a, but with a sharper cut-off than for the case of the exponential length distribution.

3 A Content-Based Approach to Dynamics of Gene Regulation

The dynamics of gene regulation networks has been modelled by Random Boolean Networks (RBN) [12, 13], where the genes at the nodes take Boolean values

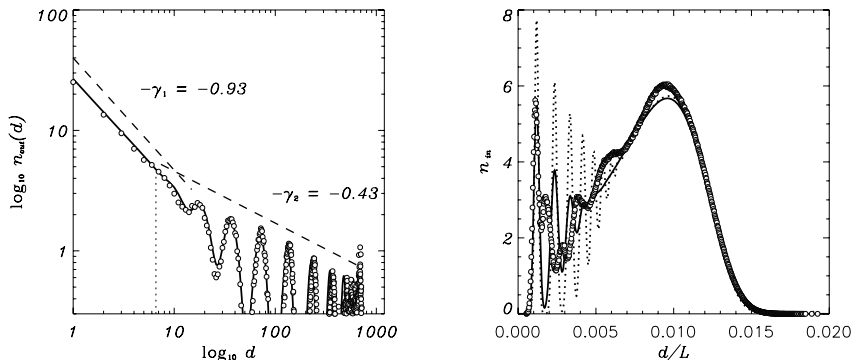


Fig. 1. The out-degree and in-degree distributions (for $L = 1.5 \times 10^4$ and $L = 4 \times 10^4$ averaged over 2×10^4 random realisations of the random string). Numerical results are indicated by circles, the solid lines are the theoretical expression. See Ref. [8].

$\sigma_i(t)$, $i = 1, \dots, N$, indicating their state of activation (“on” or “off”), determined by randomly assigned Boolean functions with inputs from other genes. The networks may be wired randomly, or with fixed in-degree. Here we present a model designed to mimic the dynamics of gene regulation, and which extends the content-based model of the preceding section.

In our extended model (see Fig.2) each gene, corresponding to a node of the regulatory network, is represented by a linear code having *two* distinct regions: *i*) a regulatory sequence (RS) through which the activation of this gene is regulated by the binding of transcription factors (TFs), and *ii*) a coding region which gets translated either into a TF or a structural protein (also see [9, 11]). The sub-sequences occurring in a RS constitute the possible “binding sites” for the TFs.

Let us denote the regulatory and coding regions of the i th gene by ρ_i and π_i , $i = 1, \dots, N$, where the lengths of the two regions, l_{ρ_i} and l_{π_i} may be chosen from different distributions. In this study we have taken the minimum length of the codes associated with the TFs, and the corresponding “binding sites,” to be $l_{\min} = 1$. Larger values may be used to tune the cut-off of the degree distribution at large degrees.

Each character in the strings ρ_i or π_i is chosen from an alphabet of r letters with uniform probability $1/r$. We will denote the m th subsequence of the string ρ_i by $\rho_i^{(m)}$, where $m \equiv (\nu, l)$ with ν specifying the first letter, and l , the length of the sub-sequence. Note $\nu = 1, \dots, l_{\rho_i}$ and $l \leq l_{\rho_i} - \nu + 1$.

An interaction between a pair of genes is now defined via a sequence-matching condition (corresponding to Eq.(2)) between the TF of the i th gene and sub-sequences in the RS of the j th gene. Defining

$$w_{ij}^{(m)} = \begin{cases} 1 & \text{if } \pi_i = \rho_j^{(m)} \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

the element of adjacency matrix between the i th and j th nodes becomes $w_{ij} = 1 - \prod_m [1 - w_{ij}^{(m)}]$, being nonzero only if at least one of the $w_{ij}^{(m)} = 1$. Note that $w_{ij} \neq w_{ji}$, in general.

The activation state of the system at time t , $\Sigma(t) \equiv (\sigma_1(t), \dots, \sigma_N(t))$, evolves under a set of Boolean operators associated with each node of the network. The input to the Boolean operators B_j are the “binding states” $b_j^{(m)}(t) = w_{ij}^{(m)} \sigma_i(t)$ of the sub-sequences $\rho_j^{(m)}$. If there exists a protein matching the sub-sequence $\rho_j^{(m)}$ in the regulatory sequence of the j th gene, *and* if the gene coding this protein is “on” at time t ($\sigma_i(t) = 1$), then it is assumed that this protein will bind the said sub-sequence, in which case $b_j^{(m)}(t) = 1$; otherwise it is zero. For those sub-sequences which have no matches, the “binding states” are set identically equal to zero.

The truth table defining the operator $B_j(\{b_j^{(m)}(t)\})$ is constructed by assigning to each set $\{b_j^{(m)}\}$ of binding states, a value 1 or 0, with probability q or $1 - q$ as illustrated in Fig.2. Note that nodes with identical RSs will have identical Boolean operators associated with them. Given an initial state of the system, $\Sigma(0)$, the dynamics of the system is now given by $\Sigma(t + 1) = (B_1(t), \dots, B_i(t), \dots, B_N(t))$.

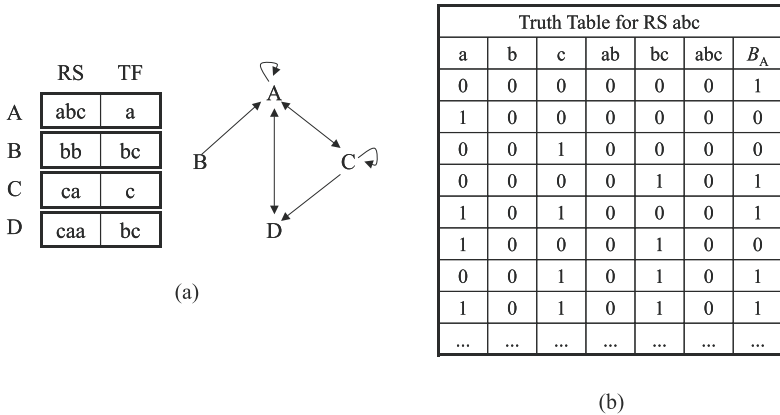


Fig. 2. A small network with four nodes (A,B,C,D) and their associated RS and TF sequences are shown in panel (a). The directed edges satisfy the matching conditions between the TFs and sub-sequences of the RSs. The truth table for the Boolean function associated with the RS of node A is shown on the left hand side of panel (b). The first six columns show the binding states of the “binding sites” a,b,...,abc. The last column gives the value B_A which has been randomly assigned to the set of binding states in that row (not all the sets are shown).

4 Simulation Results

We have simulated the dynamics on the content-based random Boolean networks defined above, with identical distributions for the string lengths associated with the RS and TF regions of each node i , choosing either (truncated) exponential

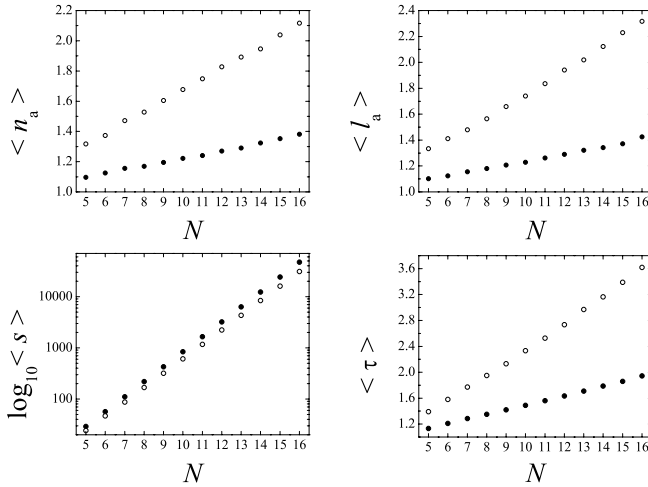


Fig. 3. Average values of the number, lengths, basin sizes and transient times for the attractors of the dynamics for Gaussian (\bullet) and exponential (\circ) string-length distributions. We have taken $r = 2$, $p = 0.5$. The distributions are truncated and normalised over the interval $[1, 25]$. The “Gaussian” distribution has $\langle l \rangle = 13$, $\sigma^2 = 50$, whereas the “exponential” distribution has a characteristic length of $\xi \simeq 10$.

or Gaussian distributions. The choice of an exponential length distribution was motivated by the fact that it lends itself to analytical treatment [8]. However, a careful analysis [11] of the length distribution of the binding sites in the regulatory regions of yeast [18, 19] reveals that it is a somewhat skewed, unimodal distribution with a mean around 15 and a variance of about 4, when expressed in binary code. We adjusted the parameters of the Gaussian distributions we used so as to give networks that are not very sparse. Work is in progress to choose different appropriate length distributions for the TFs and the short binding sites that they recognize, and the regulatory regions (RSs) in which the binding sites are embedded. [20]

For systems having a finite number N of nodes, the volume of the phase space $\Omega = 2^N$, is also finite. If we start from an initial configuration and follow its trajectory we will revisit some already visited states and fall into a cyclic orbit known as an *attractor* of the system. The mean values of n_a and l_a , the number and length of the attractors, give us some clue about the system’s stability and versatility. We find (see Fig.3) that $\langle n_a \rangle$, $\langle l_a \rangle$ as well as the average transient time $\langle \tau \rangle$ increase linearly with system size N , for both the Gaussian and exponential string length distributions. However in all cases, the exponential has higher growth rates with N for the above quantities, whereas the Gaussian length distribution gives more stable results.

It is of interest to compare our finding with those of Aldana [16] who has simulated Random Boolean dynamics on random scale-free networks, especially with regard to whether the behaviour is ordered (where differences in initial

conditions are wiped out), critical, or “chaotic,” by which we would like to understand that the lengths of the orbits grow exponentially with N . The results we get for the distribution of the number of precursors of phase points, and the distribution of the number and length of the attractors, are very similar to those found by Aldana [16] for scale-free networks with $\gamma_c = 2.5$ for $p = 1/2$, exhibiting “critical” or borderline behaviour. The scaling exponents for our content-based network in the large N limit are decidedly smaller than this value, and would rather be expected to show chaotic behaviour within Aldana’s approach. This unexpected stability of the dynamics on content-based networks also holds for the mean connectivities: For the relatively small $N \leq 16$ realisations which we have simulated, the average undirected connectivities are smaller than 2 for the truncated Gaussian, and larger than 2 for $N \geq 9$, for the exponential length distribution.

We have also simulated the average overlap function $\langle x(t) \rangle$,

$$\langle x(t) \rangle \equiv 1 - \frac{1}{N} \sum_{i=1}^N \langle |\Sigma(t) - \Sigma^{(i)}(t)| \rangle \quad , \quad (5)$$

where $\Sigma^{(i)}(0)$ differs from $\Sigma(0)$ at the i th node. The average is performed over 10^4 realisations of the network, and all $\Sigma(0)$. Following the trajectory of this quantity under successive steps of the dynamics, we find, for each value of N considered, that it converges to a set of points in a rather small but finite interval lying below unity, as shown in Fig. 4, which becomes shifted to smaller values for larger N , as both $\langle n_a \rangle$ and $\langle l_a \rangle$ grow. Even for small N , $n_a > 1$ with small but finite probability, and the phase points to which trajectories originating in different basins of attraction converge are separated by finite distances. In the presence of periodic orbits of lengths $1 \leq l_i$, one obtains a set of $\binom{M}{2}$ such finite

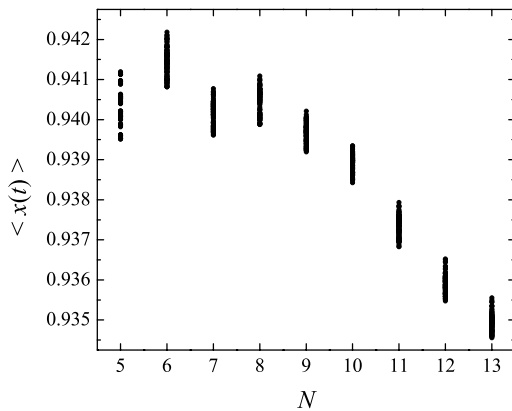


Fig. 4. The long-time trajectories of the overlap function for different network sizes $5 \leq N \leq 13$, for the exponential length distribution. Averages have been taken over 10^4 network realisations and all pairs of configurations having initial overlap $x(0) = 1 - 1/N$.

distances between pairs phase points, where $M = \sum_i^{n_a} l_i$. Thus, the persistence of distances between randomly chosen points in phase space does not automatically signal “chaotic” behaviour, but the existence of multiple and/or periodic attractors. The evolution of the $x \neq 1$ fixed point with N and its consequences for the phase diagram, [17] is under further investigation.

5 Discussion

In conclusion, let us note that the content-based network which we have proposed offers ample possibilities for the modelling of biological networks. Due to the interactions that can arise between sub-sequences of sufficiently long codes, we see that networks with topological properties similar to those observed today [6] can arise spontaneously, without the need to be assembled from scratch by evolutionary processes.

The Boolean Dynamics on our content-based network yields an increase in the number and complexity of the attractors with the number of nodes, and seems to be near-critical, although, on the basis of the Mean-Field arguments in [16] one would expect “chaotic” behaviour. We note that the expression patterns of yeast have recently been analyzed by Zivković et al. [21], and they find both periodic and stochastic behaviour, considering the differential expression levels of the genes. The discrete nature of the present model does not allow us to make detailed comparisons with their results. Nevertheless, we also find patterns where subsets of genes are going through cycles of different lengths, while some nodes are frozen in one or the other state, and that certain perturbations to the state of expression may switch the expression pattern to that of a neighbouring basin of attraction.

Acknowledgements. It is a pleasure to thank M. Mungan and A. Kabakçioğlu, with whom we have collaborated on some of the results reviewed here, for many useful discussions. AE gratefully acknowledges partial support from the Turkish Academy of Sciences.

References

1. Bollobas, B.: *Modern Graph Theory*. Springer-Verlag, New York (1998)
2. Pastor-Satorras, R., Vespignani, A.: *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, London (2004)
3. Dorogovtsev, S.N., Mendes, J.F.F.: *Evolution of Networks*. *Adv. Phys.* **51**, (2002) 1079–1187
4. Albert, R., Barabási, A.-L.: *Statistical Mechanics of Complex Networks*. *Rev. Mod. Phys.* **74** (2002) 47–97
5. Sole, R.V., Pastor-Satorras, R.: *Complex Networks in Genomics and Proteomics*. In: Bornholdt, S., Schuster, H.G. (eds.): *Handbook of Graphs and Networks*. Wiley-VCH Verlag, Berlin (2002)
6. Barabási, A.-L., Oltvai, Z.N.: *Network Biology: Understanding the Cell’s Functional Organization*. *Nat. Rev. Genet.* **5** (2004) 101–113

7. Balcan, D., Erzan, A.: Random model for RNA interference yields scale free network. *Eur. Phys. J. B* **38** (2004) 253–260
8. Mungan, M., Kabakçoğlu, A., Balcan, D., Erzan, A.: Analytical solution of a stochastic content-based network model. *J. Phys A: Math Gen.* **38** (2005) 9599–9620
9. Reil, T.: Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In: Floreano, D., Nicoud, J.D., Mondada, F., (eds.): *Advances in Artificial Life. Lecture Notes in Artificial Intelligence*, Berlin Heidelberg, Springer-Verlag (1999) 457–466
10. Alberts, B. et al.: *Molecular Biology of the Cell*. Chapter 9. Garland Science, N.Y. (2002)
11. Kabakçoğlu, A., Mungan, M., Balcan, D., Erzan, A.: in preparation
12. Kauffman, S.A.: Metabolic stability and epigenesis in randomly connected nets. *J. Theor. Biol.* **22** (1969) 437–467
13. Kauffman, S.A.: *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, N.Y. (1993)
14. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Resilience of the Internet to Random Breakdowns. *Phys. Rev. Lett.* **85** (2000) 4625–4628
15. Carlson, J. M. and Doyle, J.: Highly Optimised Tolerance: A mechanism for power laws in designed systems. *Phys. Rev. E* **60**, (1999) 1412–1427
16. Aldana, M.: Boolean dynamics of networks with scale-free topology. *Physica D* **185** (2003) 45–66
17. Derrida B., Eckmann, J.-P., Erzan, A.: Renormalisation groups with periodic and aperiodic orbits. *J. Phys. A: Math. Gen.* **16** (1983) 893–906
18. Lee, T.I. et al. : Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*. *Science* **298** (2002) 799-804
19. Harbison, C.T. et al. : Transcriptional regulatory code of a eukaryotic genome. *Nature* **431** (2004) 99-104
20. Oikonomou, Th. and Provata, A.: Non-extensive trends in the size distribution of Coding and Non-coding DNA sequences in the Human Genome. *Eur. Phys. J. B*, in press. The length distributions for the coding (non-coding) regions of the human genome are found to display different power law tails, here interpreted as indicative of short (long) range correlations, depending on the exponents. Note, however, that some of the so called intergenic non-coding regions actually code the highly conserved binding sites for the transcription factors. See Refs. [10, 19].
21. Zivković, J., Tadić, B., Wick, N., and Thurner, S. : Statistical Indicators of Collective Behaviour and Functional Clusters in Gene Networks of Yeast. *Eur. Phys. J. B*, in press.

Social Connections and Access Charges in Networks

Rodrigo Harrison¹, Gonzalo Hernandez², and Roberto Munoz³

¹ P.U.C. de Chile, Instituto de Economía

² UNAB, Escuela de Ingeniería Civil and
U. de Chile, Centro de Modelamiento Matemático

³ CIDE Mexico, División de Economía and
U.T.F.S.M. Chile, Departamento de Industrias

Abstract. In this paper we present a model where two interconnected network operators compete in linear prices in a market characterized by the existence of social connections among consumers, which are represented by a random regular graph. Assuming horizontal differentiation among operators, the customers select their network provider based on their preferences and the prices offered by the competing firms. In equilibrium the number of calls made to other agents depends on where they are located in the social network.

1 Introduction

The study of network interconnection problems has been a very active area of research over the last decade. One of the main reasons for the interest in this topic is its immediate impact in terms of regulatory policy, especially in telecommunication markets.¹ The literature has studied the case where competing networks need to be connected to their rivals in order to provide a service. In such a case, the access charge can play a key role in terms of the equilibrium arising in a competitive framework. For example, [7] developed a model where, in the presence of linear and nondiscriminatory pricing schemes to consumers, the access charge can be used by the firms as a collusive device. That is, the firms can agree to pay each other high access charges because their profits are maximized in this way. However, consumer and social surplus are reduced, consequently, regulatory intervention is called for. However, a key assumption has not been scrutinized enough. They assumed a balanced calling pattern, meaning that “the percentage of calls originated on a network and completed on the same network is equal to the fraction of consumers subscribing to this network”. In order to represent the assumption in a simple way, they assume that demand is formed by a continuous of individuals where each of them connects (or makes calls to) other individuals with the same intensity if they are confronted with the same price of the service.

¹ See [1] for a survey.

In a mobile phone market, for example, this modeling approach implies that each person, confronted to equal prices, would make the same number of calls to their relatives and friends than to any other unrelated person in the market. This unrealistic assumption is usually made for simplicity, because it permits to characterize market shares for the firms in a very simple way. More importantly, the assumption permits to characterize the equilibrium behavior of the players using closed form expressions that facilitate the comparative static. However, to the best of our knowledge, no attempt has been made to study the robustness of the results of these models to the presence of such a strong behavioral assumption.

In this paper we depart from the traditional approach by abandoning the assumption of the existence of a continuous of consumers connecting to each other with the same intensity. Instead, we present a model where two interconnected network operators compete in linear and non discriminatory prices in a market characterized by the existence of social connections among consumers, which are modeled through a social network. The balanced calling pattern is no longer an assumption, but it can arise in equilibrium.² As usual, each operator has to pay an access charge to its rival when the services demanded by its consumers have to be completed in the rival network. Assuming horizontal differentiation among operators,³ the customers select their preferred network based on their tastes and the prices offered by the competing firms. The main difference with previous studies is that we recognize that an individual in the market belongs to a social network,⁴ therefore, if he or she makes a call to an individual placed closer in the network then he or she receives a higher utility than from a call to an individual located farther away, even if the price for the service is the same. As a consequence, in equilibrium the number of calls made to others depends on where they are located in the social network.

The rest of the paper is organized as follows: In section 2 we develop the economic model, including the agent's demand, the firms' problem and the game played by the two firms. In section 3 we obtain numerical results when the social network is modeled using random regular graphs. In section 4 the conclusions are stated.

2 The Economic Model

The model assumes the existence of a social network, represented by a graph g . Nodes in the graph represent agents (indexed by $i \in I$) and the links show how people are interconnected socially. There are two firms, A and B , offering communication services (i.e. two cellular companies) and individuals have to decide which firm to subscribe to. In order to make the affiliation decision, agents take

² See [3] and [4] for results considering heterogenous agents.

³ Horizontal differentiation means that the services offered by the competing networks are not different in quality, thereby consumer preferences determine affiliation decisions.

⁴ See [5] and [6] for further details on network theory and economics.

into account the price⁵ per call offered by each firm, p_A and p_B , and his or her taste. Taste is modeled in a standard way: each agent i in the social network (i.e. each node in g) is endowed with a realization of a taste random variable x_i , from a cumulative density function F with support in $[0, 1]$. In what follows we assume that firm A is “located” in 0 and firm B in 1. None of them provide the “ideal service” to agent i , positioned in x_i (this would be the case if some network were located precisely in x_i). Introducing a unit transportation cost of t , the cost of selecting a service different from i ’s ideal one is tx_i if agent i selects network A or $t(1 - x_i)$ if network B is preferred.

2.1 The Agent Demand

Consider the affiliation decision problem of agent i . If agent i decides to connect to network $j = A, B$ then we will say that he/she belongs to the set I_j .

Agent i ’s demand for calls is represented by the vector $q_i = (q_{ij})_{j \in g, j \neq i}$, where the generic element q_{ij} is the number of calls that agent i makes to agent j . Then the gross utility of agent i can be described as follows:

$$U_i(q_i) = \sum_{j \in g, j \neq i} \lambda_i \delta^{t_{ij}} u(q_{ij}) \quad \text{with} \quad u(q_{ij}) = \frac{q_{ij}^{1-1/\eta}}{1-1/\eta} \tag{1}$$

where:

δ : is a discount factor, that is $0 < \delta < 1$.

t_{ij} : is the shortest distance (in terms of links) connecting agents i and j . The more distant is j from i in the social network, the lower is the utility of a call. If they are not connected then $t_{ij} = \infty$.

η : is a constant parameter representing the elasticity of demand.

λ_i : scale factor associated with the total amount of social connections of agent i .⁶

Now let us start assuming that i decides to connect to firm A . Given that agent i observes the price of network A , p_A , he/she is going to select the vector of calls $q_i = (q_{ij})_{j \in g, j \neq i}$ to all his contacts in the network g so that:

$$V_i(p_A) = \max_{q_i} \left\{ \lambda_i U_i(q_i) - p_A \sum_{j \in g, j \neq i} q_{ij} \right\} \tag{2}$$

Solving this maximization problem, we obtain his/her demand’s components:

$$q_{ij}(p_A) = \left(\frac{p_A}{\lambda_i \delta^{t_{ij}}} \right)^{-\eta} \tag{3}$$

⁵ We just consider linear and non discriminatory prices.

⁶ In the simulation we will use the scaling parameter $\lambda_i = (\delta C(g_i)^{\frac{1}{\eta}})^{-1}$, where $C(g_i)$ stands for the total amount of connections of the agent.

Intuitively, for the same price p_A , agent i makes more calls to contacts located closer in the social network g than farther in it. Therefore, plugging into equation 2 we get the indirect utility function:

$$V_i(p_A) = \sum_{j \in g, j \neq i} \lambda_i^\eta \delta^{\eta t_{ij}} \frac{p_A}{\eta - 1}^{1-\eta} \tag{4}$$

and an analogous result arise for firm B .

It is easy to see that each agent, given his/her taste parameter, will select the network providing him/her the highest indirect utility net of transportation costs. Agent i 's decision of affiliation will depend on his/her taste parameter value x_i in the following way: there is an x_i^* representing an indifferent value, such that if $x_i < x_i^*$ (resp. $x_i > x_i^*$) he/she will join network A (resp. B). Such x_i^* is defined by:

$$V_i(p_A) - x_i^* t = V_i(p_B) - (1 - x_i^*) t \tag{6}$$

Solving for x_i^* ,⁷ we get:

$$x_i^* = \frac{1}{2} + \sigma \frac{(p_A^{1-\eta} - p_B^{1-\eta})}{\eta - 1} \sum_{j \in g, j \neq i} \lambda_i^\eta \delta^{\eta t_{ij}} \quad \text{with} \quad \sigma = \frac{1}{2t} \tag{7}$$

2.2 The Firm's Problem

Assuming that each firm pursues the maximization of its profits and access charges are given, then firm A (resp. B) will select its price p_A (resp. p_B) such that:

$$\begin{aligned} \max_{p_A} \pi_A(p_A, p_B) = & \\ & \left\{ \sum_{i \in I_A} \left\{ \sum_{\substack{j \neq i \\ j \in I_A}} q_{ij}(p_A)(p_A - c_A^o - c_A^f) + \sum_{j \in I_B} q_{ij}(p_A)(p_A - c_A^o - a_B) - f \right\} + \right. \\ & \left. \sum_{i \in I_B} \sum_{j \in I_A} q_{ij}(p_B)(a_A - c_A^f) \right\} \end{aligned} \tag{8}$$

where:⁸

f : is the fixed cost incurred by a firm when it affiliates a new subscriber.

c_A^o : is the cost of originating a call for firm A (c_B^o is defined analogously).

c_A^f : is the cost of terminating a call for firm A (c_B^f is defined analogously).

a_A : is the price firm A charges firm B in order to terminate a call from a subscriber of B to a subscriber of A (a_B is defined analogously).

⁷ It is important to note that x_i^* does not depend on where i 's contacts are affiliated. This is not the case when networks are permitted to use different linear prices schemes for own and rival subscribers (the discriminatory case).

⁸ In what follows, when we solve an optimization problem, we assume that $g, f, c_A^o, c_B^o, c_A^f, c_B^f, \{x_i\}_{i=1}^I, t$ are all given exogenously.

Using the previous definitions, the problem for firm A becomes:

$$\begin{aligned} \max_{p_A} \pi_A(p_A, p_B) &= (p_A - c_A^o - c_A^f) p_A^{-\eta} \sum_{i \in I_A} \sum_{\substack{j \neq i \\ j \in I_A}} \delta^{\eta t_{ij}} + \\ &\quad (p_A - c_A^o - a_B) p_A^{-\eta} \sum_{i \in I_A} \sum_{j \in I_B} \delta^{\eta t_{ij}} - \\ &\quad \sum_{i \in I_A} f + (a_A - c_A^f) p_B^{-\eta} \sum_{i \in I_B} \sum_{j \in I_A} \delta^{\eta t_{ij}} \end{aligned} \tag{9}$$

The problem for firm B is analogous.

2.3 The Game

We are interested in a Subgame Perfect Nash Equilibrium (SPNE) of a two stage game played by firms. We analyze two cases.

In the first case, the unregulated case, firms are free to select noncooperatively their access charges, a_A and a_B , in a first stage and then, in the second stage, they set noncooperatively the prices to consumers p_A^* and p_B^* .

In the second case, the regulated case, the access charges for both firms are determined exogenously in the first stage (by a regulator) as equal to the marginal costs ($a_A = a_B = c_A^f = c_B^f$). In the second stage the firms set noncooperatively the prices to consumers p_A^* and p_B^* .

In both cases we are interested in the equilibrium outcome and how it is modified when some parameters change. It is also in our interest to measure consumer surplus, so given equilibrium prices (p_A^*, p_B^*) in any of the cases described above, we evaluate:

$$CS(p_A^*, p_B^*) = \sum_{i \in I_A} V_i(p_A^*) + \sum_{i \in I_B} V_i(p_B^*) - t \left[\sum_{i \in I_A} x_i + \sum_{i \in I_B} (1 - x_i) \right] \tag{10}$$

3 Numerical Results

The numerical study was performed using medium scale simulations. The social network was modeled by random regular graphs of degree d , see [2]. The access costs and prices varied within the predefined range: $[a_{\min}, a_{\max}]$, $[p_{\min}, p_{\max}]$. The goal of the simulations was to compare the outcomes of the regulated and unregulated cases. The methodology for the simulations in the unregulated case was the following:⁹

- 0) Define parameters: $I, d, \sigma, \eta, \delta, c_A^o, c_B^o, c_A^f, c_B^f$ and f
- 1) Generate random graph of degree d .
- 2) Generate random vector of network preferences: $x = (x_i)_{i \in I}$

⁹ The methodology in the regulated case is trivially implied.

- 3) For each access charge profile (a_A, a_B) in $[a_{\min}, a_{\max}] \times [a_{\min}, a_{\max}]$ we solve the second stage of the game:
 - (a) For each price profile (p_A, p_B) in $[p_{\min}, p_{\max}] \times [p_{\min}, p_{\max}]$:
 - i) Compute indifference values: $x^* = (x_i^*)_{i \in I}$
 - ii) Compute the utility functions of firms A and B :
 $\pi_A(p_A, p_B), \pi_B(p_A, p_B)$
 - (b) Compute price response functions:
 $r_A(p_A, p_B), r_B(p_A, p_B)$
 - (c) Compute SPNE price reaction functions in the second stage:
 $p_A^*(a_A, a_B), p_B^*(a_A, a_B)$
- 4) Going back to the first stage, compute access cost response functions
 $r_{a_A}(a_A, a_B), r_{a_B}(a_A, a_B)$
- 5) Compute SPNE access charges in the first stage a_A^*, a_B^* and evaluate price reaction functions at these values: $p_A^*(a_A^*, a_B^*), p_B^*(a_A^*, a_B^*)$
- 6) Evaluate indifference values at the equilibrium outcome:
 $x^*(a_A^*, a_B^*) = (x_i^*(p_A^*(a_A^*, a_B^*), p_B^*(a_A^*, a_B^*)))_{i \in I}$
- 7) Evaluate firms' benefits equilibrium outcome:
 $\pi_A(p_A^*(a_A^*, a_B^*), p_B^*(a_A^*, a_B^*)), \pi_B(p_A^*(a_A^*, a_B^*), p_B^*(a_A^*, a_B^*))$
- 8) Evaluate consumer surplus $CS(p_A^*(a_A^*, a_B^*), p_B^*(a_A^*, a_B^*))$ as in eq. 10
- 9) Repeat steps (1) to (8) for T different graphs keeping parameters defined in step (0).

The size of the social network was fixed at $I = 1024$ and random regular graphs with degrees $d = 10, 15, 20$ were used. The access charge discretization selected 5 values: 2 above and 2 below the regulated case. The price discretization considered 20 equidistant values. Table 1 below contains average results for simulations. The rest of the parameters were fixed as follows: $\sigma = 1, \eta = 1.5, \delta = 0.9, c_A^o = c_B^o = 0.75, c_A^f = c_B^f = 0.75$ and $f = 0.5$.

Table 1. Numerical results when d is modified

	d	Firm A			Firm B		
		regulated	unreg.	change %	regulated	unreg.	change %
access charge	10	0.75	1.05	40	0.75	1.05	40
	15	0.75	1.05	40	0.75	1.05	40
	20	0.75	1.05	40	0.75	1.05	40
prices	10	3.10	3.45	11.3	3.15	3.50	11.1
	15	3.05	3.40	11.5	3.10	3.20	3.2
	20	3.05	3.40	11.5	3.10	3.20	3.2
utilities	10	77958	83428	7.1	77223	83010	7.5
	12	82417	83439	1.2	81931	87951	7.4
	20	84905	85892	1.1	84400	90658	7.4

	d	regulated	unregulated	change %
consumer surplus	10	709771	672630	-5.2
	15	765856	742197	-3.0
	20	788867	764515	-3.0

4 Conclusions

Our preliminary results show that an eventual deregulation in the determination of access charges would have a big impact on the level of access charges but a relatively minor impact on prices, utilities and consumer surplus. On the other hand, an increase in the connectivity of the social network (implying an increase in demand keeping constant the number of individuals forming the network) has a minor impact over access charges and prices, *ceteris paribus* the regulatory environment. However, firms' benefits and consumer surplus increase significantly, especially in the regulated and unregulated case respectively, closing the gap between both regulatory frameworks. Overall the simulations show that deregulation can be an attractive alternative in front of a costly regulation, especially when the social network presents a high level of connectivity. This result contrasts with the standard recommendations in the literature for the interconnection problem under linear and nondiscriminatory pricing schemes.

References

1. Armstrong, M.: The Theory of Access Pricing and Interconnection, Handbook of Telecommunications Economics, vol. 1 (2002), Edited by Martin Cave et al. Amsterdam: North-Holland.
2. Bollobas, B., Random Graphs, Cambridge University Press (2001).
3. Dessein, W.: Network Competition with Heterogeneous Customers and Calling Patterns, Information Economics and Policy, vol. 16 (2004).
4. Hahn, J.: Network Competition and Interconnection with Heterogeneous Subscribers, International Journal of Industrial Organization vol. 22 (2004).
5. Jackson, M.: The Stability and Efficiency of Economic and Social Networks, mimeo (2001) Caltech.
6. Jackson, M. and A. Wolinsky: A Strategic Model of Social and Economic Networks, Journal of Economic Theory, vol. 71 (1996).
7. Laffont, J., P. Rey and J. Tirole: Network Competition: I. Overview and Nondiscriminatory Pricing, RAND Journal of Economics, vol. 29, (1998).
8. Laffont, J., P. Rey and J. Tirole: Network Competition: II. Price Discrimination, RAND Journal of Economics, vol. 29 (1998).

Topology of Cell-Aggregated Planar Graphs

Milovan Šuvakov and Bosiljka Tadić

Department for Theoretical Physics, Jožef Stefan Institute,
Box 3000, 1001 Ljubljana, Slovenia
Milovan.Suvakov@ijs.si, Bosiljka.Tadic@ijs.si
<http://www-f1.ijs.si/~tadic/>

Abstract. We present new algorithm for growth of non-clustered planar graphs by aggregation of cells with given distribution of size and constraint of connectivity $k = 3$ per node. The emergent graph structures are controlled by two parameters—chemical potential of the cell aggregation and the width of the cell size distribution. We compute several statistical properties of these graphs—fractal dimension of the perimeter, distribution of shortest paths between pairs of nodes and topological betweenness of nodes and links. We show how these topological properties depend on the control parameters of the aggregation process and discuss their relevance for the conduction of current in self-assembled nanopatterns.

1 Introduction

In recent years increased interests in various networks realizations [1, 2] revealed that several new types of graphs termed *structured graphs* are more appropriate mathematical objects to describe complex network's geometry than traditional *random graphs* [3]. The variety of structures was found to emerge through evolution processes in which nodes and links are added sequentially according to specified rules, in particular, the preferential attachment rules lead to strongly inhomogeneous *scale-free graphs* [1]. In contrast to the evolving networks, which comprise a class of *causal graphs*, the class of *homogeneous graphs* consists of graphs with fixed number of nodes and fluctuating or rewiring links according to given rules or certain optimization processes. Complex graph structures may emerge in this procedures, especially when certain global or local optimization constraints are imposed [4].

Planar graphs are special class of graphs that can be embedded in a Euclidean plane. A graph is planar *iff it does not contain a subdivision of K_5 (5-clique) and $K_{3,3}$ (minimal non-planar graph with 6 nodes)* [3]. Consequently, planar graphs fulfill Euler's law: $N_p + N = E + 1$, which is relating the number of nodes N , links E and polygons N_p .

In this work we suggest a new method for growing a planar cellular graph by attachment of objects—cells (polygons) of length n_p , which are chosen from a given distribution $f(n_p)$. The polygons are added sequentially in time starting from an initial polygon. In addition, we strictly impose the constraint on number

of links per node $k = 3$, which is thus fulfilled everywhere in the interior of the graph and on some nodes on the graph boundary. The attachment of cells is controlled by two parameters—the width of the distribution of cell sizes μ_2 and the parameter ν that plays the role of chemical potential of cell aggregation. In the limit of vanishing attachment potential $\nu \rightarrow 0$ the growth process resembles the one in diffusion-limited aggregation [6]. However, aggregated are spatially extended cells of particles rather than single particles.

Emergent structures of cellular networks are resembling of soap froths [7] or patterns of nano-particles self-assembled through nonlinear dynamic processes [8, 9]. Typically, a pattern of cells appears when nano-particles are immersed in a liquid film, which is then allowed to evaporate until holes of different sizes open-up leaving particles in the walls between the holes [9, 10]. Generally, the structure of the patterns effects the physical processes on them, such as current transport [11]. It is therefore important to understand the topology of the aggregated cellular networks in detail. Here we study the topological properties, such as shortest paths between nodes, topological centrality, and fractality of the graph’s perimeter in different cellular networks obtained by varying the control parameters of the aggregation processes.

2 Cell Aggregation

The basic idea is to make growing model of planar graph with given distribution of cell (polygon) sizes $f(n_p)$ and with degree of nodes $2 \leq k \leq 3$.

2.1 Topological Constraint

For this purpose we observe some topological constraints on the distribution of cell sizes: (i) $f(n_p)$ is defined for $n_p \geq 3$, for non-clustered graph we fix $f(3) = 0$; (ii) planar graph obeys Euler’s law: $N_p + N = E + 1$. Among these the homogeneous plane-filling structures are of special interest [7]. For this class of graphs majority of nodes are in the interior of the graph, ie., nodes with degree $k = 3$. Therefor $3N \approx 2E$ and Euler’s law becomes

$$6N_p = 2E + 6. \tag{1}$$

For large system with distribution of cell sizes $f(n_p)$ we have

$$N = N_p \sum_{n_p} \frac{n_p f(n_p)}{3}, \quad E = N_p \sum_{n_p} \frac{n_p f(n_p)}{2}. \tag{2}$$

Substituting second relation into (1): $6N_p = N_p \sum_{n_p} n_p f(n_p) + 6$, then for large $N_p \gg 1$ one can find that the average cell size is equal to six

$$\langle n_p \rangle \equiv \sum_{n_p} n_p f(n_p) = 6. \tag{3}$$

We use lognormal distribution of polygon size

$$f(n_p) = \frac{1}{s\sqrt{2\pi}x} e^{-\frac{\ln^2 x/x_0}{2s^2}}, \tag{4}$$

which is most often found in experiments [9]. Using the condition (3) the number of independent parameters in (4) is reduced

$$\langle n_p \rangle = 6 \quad \Rightarrow \quad x_0 = 6e^{-s^2/2}, \quad s^2 = \ln\left(1 + \frac{\mu_2}{36}\right), \tag{5}$$

where second central moment μ_2 remains as the control parameter in our case.

2.2 Model

Starting from an initial cell, at each time step a new cell with size taken from the distribution $f(n_p)$ is attached to the graph boundary of the graph according to the following rules:

1. In the aggregation only nodes with degree 3 can be closed inside the graph;
2. New link can be assigned only to nodes with degree 2;
3. Potential nesting place is searched as an array of nodes on graph boundary with degree 3 limited with two nodes of degree 2. The nested part of the cell is identified with the nodes of the nesting string. Therefore, number of extra nodes to be added is $n = n_p - l$, where l is length of the nesting string. We select the nesting place with probability $p \sim e^{-\nu n}$, where the parameter ν plays the role of the chemical potential for addition of new nodes.

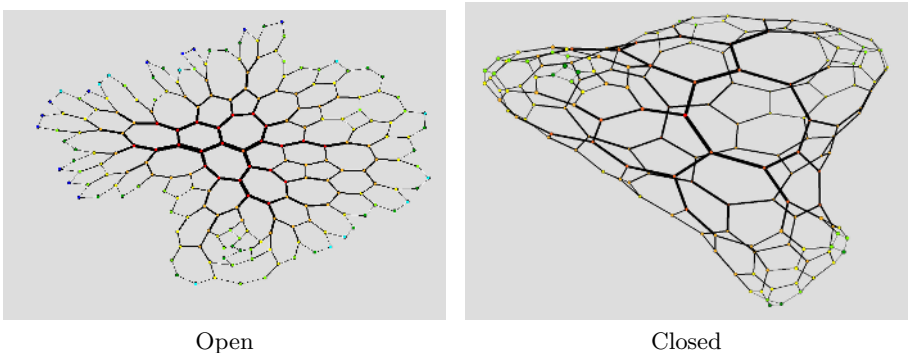


Fig. 1. Two possible types of cell-aggregated *planar graphs*: open and closed structures, obtained by lognormal distribution with $\mu_2 = 1.0$, and aggregation potential $\nu = 5.0$

We implemented this algorithm in *C++* according to following steps:

```

Initial graph: one polygon of size np taken from f(np)
For i=2 to Np
  np = next random from distribution f(np)
  If(there is no nodes on graph boundary with degree 2) exit(1)
  For all j=(periphery node with degree 2)
    d=distance to the next node on graph boundary with degree 2
    Number of new nodes n = np-d-1
    If(n>0) p(j)=exp(-nu*n)
  End of loop j
  If(there is no j with n>0) exit(2)
  Normalize p(j)
  j = next random from distribution p(j)
  Add new polygon with size np linked with
    node j and next node on graph boundary with degree 2
End of loop i
exit(0)

```

Depending on model parameters of the growth process and its stochasticity three possible exit cases are:

- exit(0) - Open structure (planar graph with N_p polygons);
- exit(1) - Closed structure (after some number of step there are no more nodes of degree 2 and structure stops to grow, no nesting places of any size);
- exit(2) - No nesting place available for current cell. In this case one can take next cell, which in turn perturbs the actual distribution.

We never experienced the exit(2) situation for the range of parameters $\mu_2 \in [0.5, 2.0]$, $\nu \in [0, 5]$ and $N_p = 1000$ in huge number of samples. Two examples of the emergent open and closed structures are shown on Fig. 1. More examples of cellular networks are shown in Fig. 2 for varying parameters μ_2 and ν .

3 Fractal Dimension of Network Perimeter

During the nesting growing process in one step number of nodes N increases by $n < n_p$. Number of added nodes n at each step depends on the cell size and length of the nesting string. Therefore, $N \approx \kappa N_p$ where $\kappa = \langle n \rangle$ is average growth rate. For open structures (see Fig. 1-2) boundary of the graph becomes fractal, depending on the control parameters. In fact, length of the graph boundary L grows as a power of the number of cells N_p (or network size N) with fractal dimension D defined by $L \sim N_p^D$.

In Fig. 3 we show how number of nodes on the graph boundary increases with N_p . Each point is averaged over 10 emergent growing networks. The dimension D is in the range $\frac{1}{2} \leq D \leq 1$, when $D = 1$ correspond to structures of high fractality, that is obtained for small values of the parameter ν . $D = \frac{1}{2}$ correspond to planar "circle like" structures with reduced fractality. For $\mu_2 > 0$ we observe a

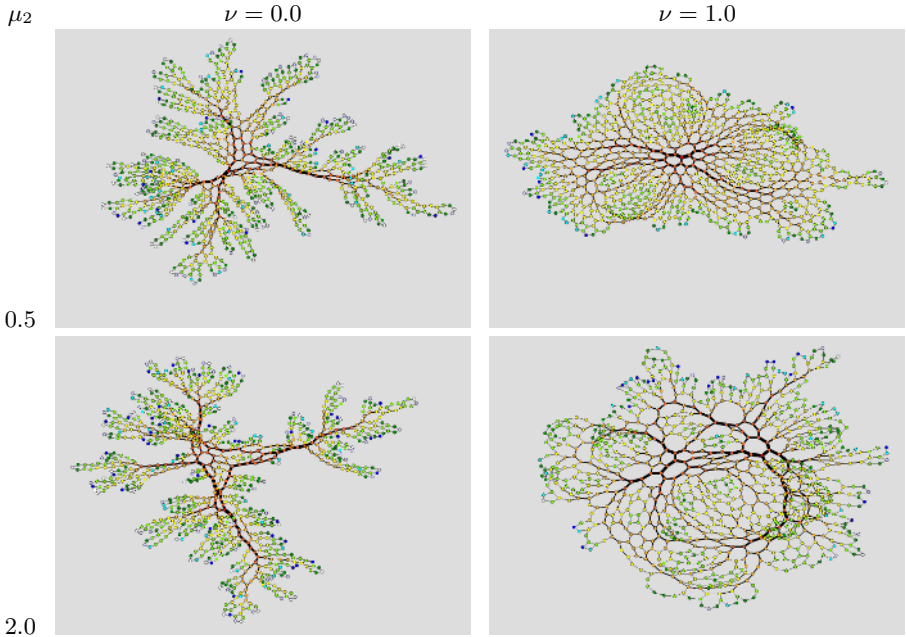


Fig. 2. Cell-aggregated planar graphs with lognormal distribution of polygon size for various values of width μ_2 and cell aggregation potential ν . Width of lines represent topological betweenness (centrality) of links calculated in Section 4.2.

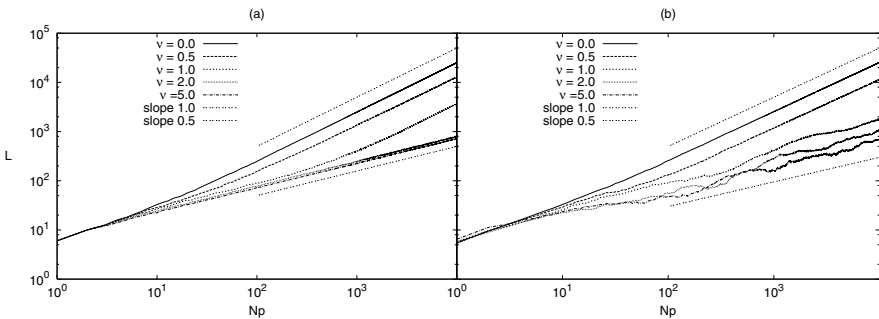


Fig. 3. Scaling of the network perimeter for (a) $\mu_2 = 0$ hexagons only (b) $\mu_2 = 2.0$

continuous crossover between these two limits (see Fig. 3b for $\mu_2 = 2$). However, in structures with homogeneous cell distribution ($\mu_2 = 0$ - hexagons only) a sharp transition seems to occur at $\nu_c \approx 1.5$.

4 Shortest Paths and Centrality on Cellular Networks

In this section we consider global topological properties of the cell-aggregated planar graphs and their dependence on the control parameters μ_2 and ν .

4.1 Shortest Paths

Shortest path between two nodes is defined as path along the smallest number of intermediate links [3]. We implemented an algorithm for counting shortest paths of Dijkstra type [12]. In Fig. 4 we show distribution of lengths of shortest paths between all pairs of nodes on network. All networks are for fixed $\mu_2 = 1.0$ and approximately of the same size $N \approx 1000$ nodes. Each point in Fig. 4 is averaged over 100 sample networks. We found similar results for other μ_2 values.

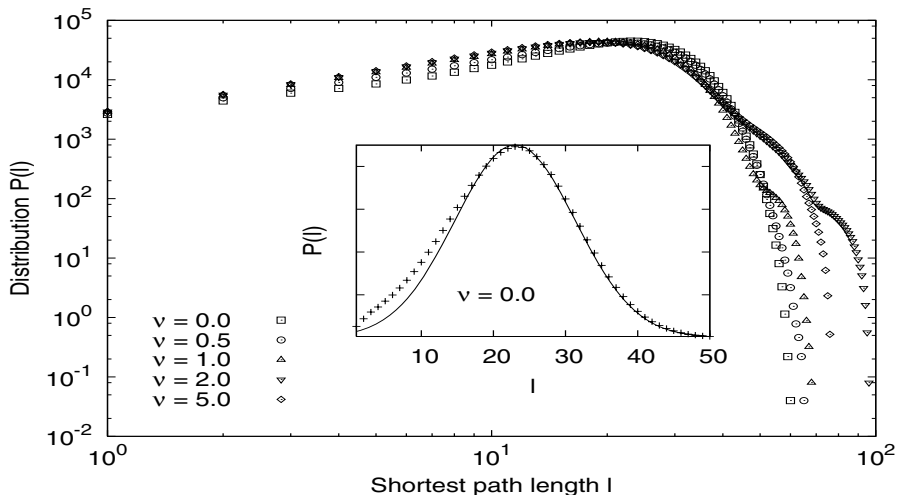


Fig. 4. Distribution of lengths of shortest paths on networks for fixed $\mu_2 = 1.0$ and various values of ν . Inset: The distribution in the case $\nu = 0$ is shown on linear scale. Solid line: Gaussian with $l_0 = 23$ and $\sigma = 8.37$.

All these networks have similar topology at local level, because the number of links at all interior nodes is constant $k = 3$. Therefore, distributions of shortest distances at small scale are similar for all values of parameter ν . Differences in global topology appear on large scale for lengths larger than peak value $l_0 \sim 25$, which manifest in occurrence of additional peaks (see Fig. 4). The probability of long paths increases for larger values of the parameter ν . Whereas in the limit $\nu = 0$ the distribution of length of shortest paths on large scale can be approximated with a normal distribution (inset on Fig. 4).

4.2 Centrality Measures

Betweenness centrality of a node in network is defined by [13, 3]

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{6}$$

where σ_{st} is total number of shortest paths between nodes s and t , and $\sigma_{st}(v)$ is number of these paths that node v lies on. Betweenness of links is defined in

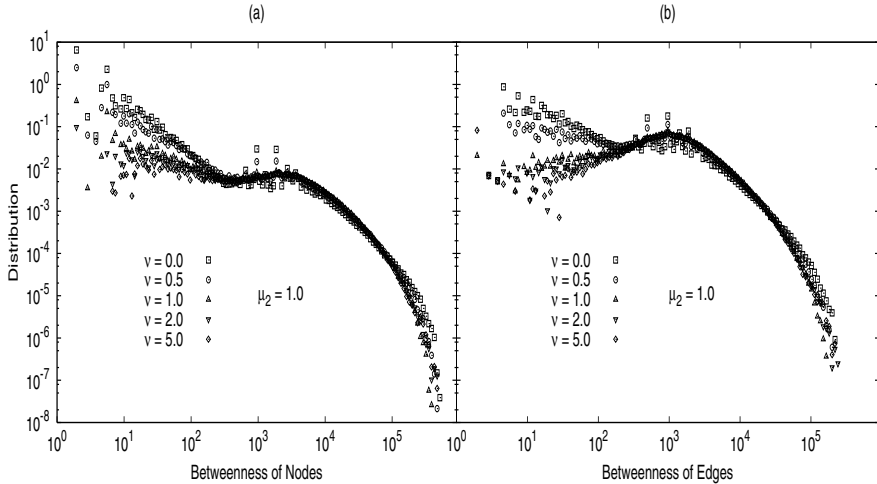


Fig. 5. Distribution of betweenness (a) of nodes (b) of links

analogous way. In our algorithm, we record number of shortest paths through each node and through each link on a network. In Fig. 5 we show distributions of betweenness of nodes and links averaged over 100 sample networks with fixed $\mu_2 = 1.0$ and size $N \approx 1000$ nodes. For inhomogeneous networks, distributions of these two betweenness measures can be substantially different. In our case, however, they are similar because all interior nodes have fixed degree $k = 3$. We find that distributions at small scale strongly depend on parameter ν , which results from the graph boundary. Similar conclusions hold for other μ_2 values.

In Fig. 2 thick lines represent links with large betweenness. For this type of networks the strongest lines, which make the skeleton of the graph, are connecting the nodes with largest centrality measure.

5 Conclusions

We have introduced a new algorithm for growth of graphs by aggregation process of extended objects - polygons with size distribution. Depending on aggregation conditions, which are determined by two parameters controlling the distribution width and attachment potential, we can get a wide spectrum of emergent structures. In this paper we presented some results for the case of lognormal distribution of cells and additional constraints, leading to the emergent non-clustered planar graphs with a constant node connectivity. The algorithm works for variety of cell distributions and constraints, that may result in diverse opened (fractal) or closed structures. For instance, for a special set of parameters we can get closed structures of C_{60} type.

We measured several topological properties of these networks in quantitative details—fractality of the graph boundary, shortest paths, and betweenness

centrality. These properties are important for some dynamical process on networks such as electrical conductivity [11] via single electron tunneling [9, 14].

Acknowledgments. M.S. thanks financial support from the Marie Curie Research and Training Network MRTN-CT-2004-005728 project. B.T. is supported by the program P1-0044 of the Ministry of high education, science and technology (Slovenia).

References

1. Dorogovtsev, S. N., and Mendes, J. F. F.: *Evolution of Networks*. Oxford University Press 2003
2. Boccaletti S., Latora V. *et al.*: *Complex Networks: Structure and Dynamics*. Physics Reports **424** (2006) 175-308
3. Bollobás, B.: *Modern Graph Theory*. Springer (New York) 1998.
4. Biely C., and Thurner S.: *Statistical Mechanics of Scale Free Networks at a Critical Point: Complexity Without Irreversibility*. cond-mat/0506140.
5. Newman, M. E. J.: **The structure and function of complex networks**. SIAM Rev. **46** (2003) 167-225
6. Witten, T. A., and Sander L.M.: *Diffusion-Limited Aggregation, a Critical Phenomenon*. Phys. Rev. Lett. **47** (1981) 1400-1403
7. Stavans, J.: *Evolution of cellular structures*. Rep. Prog. Phys. **56** (1993) 733-789
8. B. Tadić, *From Microscopic Rules to Emergent Cooperativity in Large-Scale Patterns*, in **Systems Self-Assembly: multidisciplinary snapshots**, N. Krasnogor, S. Gustafson, D. Pelta, J. L. Verdegay (Eds.) Elsevier(2005).
9. Moriarty, P., Taylor M. D. R., and Brust, M.: *Nanostructured Cellular Networks*. Phys. Rev. Lett. **89** (2002) 248303
10. Moriarty, P. : *Nanostructured materials*. Rep. Prog. Phys. **64** (2001) 297-381
11. M. Šuvakov *et al.* (in preparation)
12. Ahuja, R. K., Magnanti, T. L., and Orlin, J. B.: *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall (New Jersey) 1993
13. Freeman, L.C.: *A set of measures of centrality based on betweenness*. Sociometry **40** (1977) 35-41
14. Ferry, D. K. and Goodnick, S. M. *Transport in Nanostructures*. Cambridge University Press 1997

Geographical Construction of Scale-Free Networks with Both Short Path Lengths and Hops

Yukio Hayashi¹ and Jun Matsukubo²

¹ Japan Advanced Institute of Science and Technology, Ishikawa, Japan

² Kitakyusyu National College of Technology, Fukuoka, Japan

Abstract. We find the structural effect in geographical networks on the optimal paths and on the robustness of the connectivity. The communication efficiency are measured by the average path lengths and hops in the typical planar networks: Delaunay triangulation, random Apollonian network, and our proposed model with the well-balanced properties. The dynamic configuration will be useful especially for ad hoc communication.

1 Introduction

Complex networks have been studied with great interest inspired from physics to biology, computer science, and other fields, since the surprisingly common topological structure called *small-world* (SW) or *scale-free* (SF) has been found in many real systems [1]. It has good properties in the meanings of economical and efficient communication by small number of hops in a connected network with a few links [2] and of the robustness against failure [3]. Moreover, the restriction of link lengths has been observed, e.g. Internet at both router and AS levels [4], road networks, and flight-connection in a major airline [5]. Indeed, the distribution of link lengths was inversely proportional to the lengths [4] or exponentially decayed [6]. In other words, the length is not equal but various in a space, therefore some long-range links are more dominant to the shortest path whose distance is defined by a sum of the link lengths on the path between two nodes. This situation called disorder has been recently studied in complex networks [7],[8]. On the other hand, although efficient transport of packets, passengers, supplies, or other quantities depends on both the topology and routing methods [9], there are local rules to find the optimal paths in a special class of networks such as planar graphs.

In this paper, we consider geographical SF network models for a number of research fields including urban planning, electric circuits, distributed robots, sensor networks, communication networks, and so on. In particular, we discuss dynamic configuration of planar networks for ad hoc communication. The planarity is important to avoid interference of the wireless beam, or to construct communication lines on the surface of earth.

In the state-of-the-art studies, a few geographical SF network models have been known with theoretical analyses in the evolutionary mechanisms of power-law behavior. They are categorized into three classes: the modulated Barabási-Albert (BA) model by a penalty of the distance for connecting nodes [10],[11], SF

networks embedded in lattices [12],[13], and space-filing networks [14],[15],[16]. A brief review of these models has been reported [6]. Unfortunately, crossing of links exist except for the third class. On the other hand, as a typical model in the third class, a random Apollonian network (RA) has some long-range links which cause dissipation of the beam power or the construction cost of links, although it is based on planar triangulation without crossing of links. Thus, to reduce long-range links, we consider a modification of RA preserving the good properties of SF structure on a planar space. We investigate the shortest paths with weak disorder and find the universal scaling laws. Moreover, the robustness against the attacks on hubs is improved.

2 Models of Ad Hoc Networks

2.1 Planar Triangulation

Planar triangulation is a mathematical abstraction of ad hoc networks, in which the positions of nodes are temporarily fixed as base stations of backbone networks. Thus, the mobility of node is out of our scope to simplify the discussion. On such graphs, online routing algorithms [17] that guarantee delivery of messages using only local information about positions of the source, destination, and the adjacent nodes to a current node in the routing have been developed. To find a path through exploration is required in many cases, since knowledge about the environment in which routing takes place is not available beforehand, especially in dynamic configurations with evolution. In any case, the optimal path depends on both topological and spatial network structures, as measures of the communication efficiency, the number of hops for transfer of a message and the path length in the Euclidean distance are crucial.

On the other hand, Delaunay triangulation (DT), which is the dual of a Voronoi diagram, is the optimal planar triangulation in some geometric criteria [18] with respect to the maximin angle and the minimax circumcircle of triangle on a two-dimensional space, and widely used in practical applications for facility locations and computer graphics [19]. It is well-known as a good property that the shortest path length between any two nodes on a Delaunay graph is of the same order as the direct Euclidean distance, since the ratio of the path length to the direct distance is bounded by a constant [20]. However, the average number of hops on that graph is unknown. One of the fundamental techniques for equipping such properties is diagonal flipping. In the Delaunay triangulation, diagonal flips are globally applied to the triangles until the minimum angle of triangle is not increased by the exchange of diagonal links in a quadrilateral. Such global process is unsuitable for ad hoc networks. In contrast, RA can be constructed by local procedures for the subdivision of a randomly chosen triangle, although it has some long-range links. Thus, we compare the communication efficiency measured by the average path length and hops in the typical network models based on planar triangulation: DT in computer science, RA in complex network science, and our modification to bridge them.

2.2 Delaunay-Like SF Network

We briefly explain a random Apollonian network (RA) [14] constructed from an initial triangulation of a polygon as follows. At each time step, a triangle is randomly chosen, and a new node is added inside the triangle and linked to its three nodes. We assume the new node is set at the barycenter of the chosen triangle. The topological properties of power-law degree distribution, large clustering coefficient, disassortative degree-degree correlation, and the average small number of the minimum hops on paths between any two nodes have been theoretically and numerically analyzed [14],[15],[16]. Although RAs have the several advanced SF properties and the SW effect with a small diameter of graph, some long-range links naturally appear near the boundary edges. To reduce the long-range links, we propose a modified model from RAs. The main idea is based on a strategy for connecting nodes in distances as short as possible by adding with the diagonal flips in DTs. The proposed network is grown as follows.

Step 0: Set an initial planar triangulation in a space.

Step 1: Select a triangle at random and add a new node at the barycenter.

Then, connect the new node to its three nodes. Moreover, by iteratively applying diagonal flips, connect it to the nearest node (or more than one of the neighbor nodes) within a radius defined by the distance between the new node and the nearest node of the chosen triangle.

Step 2: The above process is repeated until the required size N is reached.

We have two variations with one nearest node and all neighbors in the local circle, whereas the diagonal flips are globally applied in DTs. Note that these nodes are limited to the connected ones by applying iterative diagonal flips. We call our model RA+NN(one/all) that means the combination with the triangulation in RAs and the rewiring to the one or all Nearest Neighbors denoted in the parentheses.

Fig. 1 illustrates the linking procedures by iterative diagonal flips: in a quadrilateral that consists of the shaded triangles, the long-range (cross) link is diagonally exchanged to the red link for maximizing the minimum angle of triangle. The dashed lines are new links from the barycenter, and form new five triangles with contours in the left of Fig. 1; The intersected black solid links with dashed ones are removed after the 2nd flips. Thus, the concentrate of links to hubs is

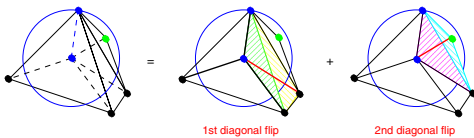


Fig. 1. Linking procedures in a Delaunay-like SF network. The intersected lines are exclusive in each shaded quadrilateral. The green node is the nearest from the barycenter.

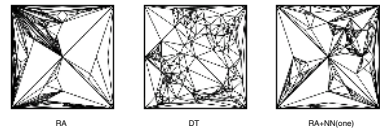


Fig. 2. Examples of RA: random Apollonian network, DT: Delaunay triangulation, and RA+NN: our proposed model

relaxed from connecting star-like stubs. Fig. 2 shows the topological characteristic that our RA+NN(one)s have the intermediate structure between those of RAs and DTs. The case of all neighbors is the same with the one's property.

3 Efficiency for Communication

3.1 Approximative Analysis of the Degree Distribution

The degree distribution is one of the important statistical characteristics related to the SF properties. We approximately derive the exponential decaying in the tail of degree distribution for the proposed network model. The occurrence of cutoffs also observed in real networks [21] is rather natural from the constraint on addition of new links.

When some links are removed from a node by multiple diagonal flips as shown in Fig. 1, the dynamical equation of the number of nodes $n(k, N)$ with degree k at the size N is given by

$$n(k+1, N+1) = \frac{k}{N_{\Delta}}n(k, N) + \left(1 - \frac{k+1}{N_{\Delta}}\right)n(k+1, N) - a\frac{k}{N_{\Delta}}n(k+1, N),$$

where N_{Δ} and a denotes the number of triangles and the average rate of the multiple diagonal flips, respectively. The 1st and 2nd terms in the r.h.s correspond to the preferential attachment through random selection of a triangle, and the 3rd term is the statistical rewiring effect by multiple diagonal flips. Note that there is no other reason for decreasing the degree. We neglect the other effects such as additional links to nodes with low degrees, because we focus on the tail of degree distribution.

By using $P(k) = n(k, N)/N$, we have

$$\frac{N_{\Delta} + N}{N}P(k+1) + k(P(k+1) - P(k)) + akP(k+1) = 0.$$

From the continuous approximation $dp/dk \approx P(k+1) - P(k)$ and $\gamma \stackrel{\text{def}}{=} (N_{\Delta} + N)/N$, it is rewritten as

$$k\frac{dp}{dk} = -(\gamma + ak)p.$$

Thus, we obtain the solution $p(k) \sim k^{-\gamma} \exp(-ak)$ for large N . Note that in RAs a power law with the exponent $\gamma_{RA} \approx 3$ has been similarly derived [14].

Fig. 3(a) shows the agreement of our approximation with the observed degree distribution. The cumulative rate a is numerically estimated by the average convergent value as shown in Fig. 3(b). Note that the time step t is equivalent to the network size N , because a new node is added at each time. Each network model is investigated in the averaging of 100 random realizations at the size $N = 1,000$ generated from the initial triangulation of a square graph.

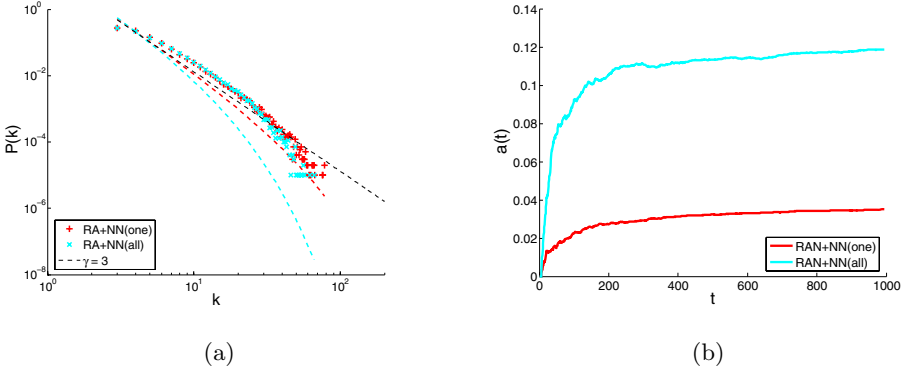


Fig. 3. Numerical estimation of (a) the power-law degree distribution with exponential cutoff. The red and cyan dashed lines show the approximations in the form $k^{-\gamma} \exp(-ak)$ with $\gamma = (N_{\Delta} + N)/N \approx 2.994$, $a = 0.03$ for RA+NN(one), and $a = 0.12$ for RA+NN(all). The black dashed line guides the slope of 3. (b) the cumulative rate $a(t)$ of diagonal flips as a function of time step t .

3.2 Weak Disorder in the Distribution of Link Lengths

In the studies of the optimal path in disordered complex networks [7],[8], each link length is associated with a weight assumed by $\exp(\delta\varepsilon)$, where the parameter δ controls the strength of disorder, and ε is a random number taken from a uniform distribution between 0 and 1. As a network approaches the strong disorder limit at $\delta \rightarrow \infty$, only the longest link becomes dominant in the shortest path length defined by the smallest sum of link lengths on a path between two nodes. At the limit, the scaling relations of the average shortest path length $\langle D \rangle \sim N^{1/3}$ for $\gamma > 4$ and $\langle D \rangle \sim N^{(\gamma-3)/(\gamma-1)}$ for $3 < \gamma \leq 4$ has been theoretically predicted [7] from the percolation on SF networks [22]. Although the relation is unknown for $2 < \gamma \leq 3$ because of the singularity in the analysis at $\gamma = 3$, $\langle D \rangle \sim (\ln N)^{\gamma-1}$ has been also numerically suggested [7].

However, the assumption of length distribution may be violated on a geometric space, in addition the strong disorder limit is an extreme case. Thus, to investigate the strength of disorder in RAs, DTs, and the proposed networks, we compare the length distributions. Fig. 4 shows the distribution $P(l_{ij})$ of link length l_{ij} in each network. The dashed lines with an equal gap from top to bottom are corresponded to the distributions of weight $2 \exp(\delta\varepsilon) / \exp(\delta)$ for $\delta = 1, 2, 4, 8, 16$, respectively. The factor $2 / \exp(\delta)$ is due to the normalization for the maximum length of the boundary edge. We find that RAs and RA+NN(one/all)s have weak disorder with small δ [8], while DTs have a slightly broad range of disorder as similar to the exponential decay in the domestic airline flight-connection [6].

3.3 Shortest Path Lengths and Minimum Hops

We investigate the average distance of path length $\langle D \rangle$ on the shortest paths, the distance $\langle D' \rangle$ on the paths of the minimum hops, the average number of hops

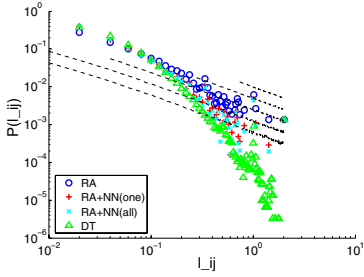


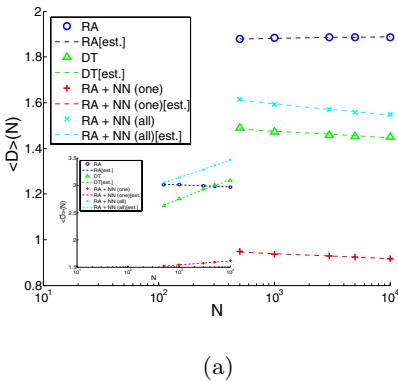
Fig. 4. The distribution of link lengths with weak disorder

Table 1. Estimated values of the exponents in the forms $\langle D \rangle \sim (\ln N)^{\beta_d}$, $\langle D' \rangle \sim (\ln N)^{\beta_{d'}}$, $\langle L \rangle \sim (\ln N)^{\beta_l}$, $\langle L' \rangle \sim (\ln N)^{\beta_{l'}}$, $\langle L \rangle \sim N^{\alpha_l}$, $\langle L' \rangle \sim N^{\alpha_{l'}}$, by the mean-square-error method for each network

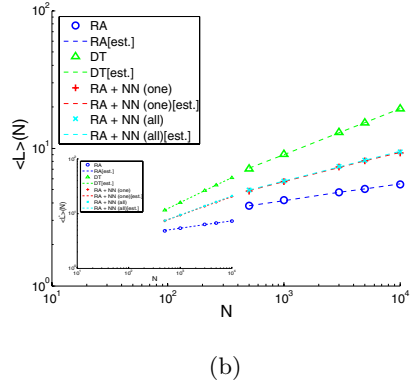
network model	β_d	$\beta_{d'}$	α_l	$\alpha_{l'}$	β_l	$\beta_{l'}$
RA	0.012	-0.039	0.121	0.136	0.920	1.036
DT	-0.068	0.416	0.332	0.455	2.525	3.452
RA+NN(one)	-0.080	0.151	0.213	0.341	1.622	2.587
RA+NN(all)	-0.106	0.320	0.216	0.346	1.641	2.628

$\langle L \rangle$ on these paths, and the number of hops $\langle L' \rangle$ on the shortest paths between any two nodes. Figs. 5(a)(b) show the SW effect. Note that the shortest path and the path of the minimum hops may be distinct, these measures are related to the link cost or delay and the load for transfer of a message. It is better to shorten both the distance and the number of hops, however their constraints are generally conflicted.

As in Table 1, we find the scaling relations estimated by straight lines for the sizes $N = 500, 1,000, 3,000, 5,000$ and $10,000$. We remark that the values of β_d and $\beta_{d'}$ differ from $\gamma - 1 \approx 2$ numerically suggested at the strong disorder limit [7], although the values of β_l and $\beta_{l'}$ are relatively close to it. The weak disorder may affect these differences. In addition, the values of α_l and $\alpha_{l'}$ are close to $1/3$ predicted at the limit [7] for the Erdős-Rényi (ER) model as the classical random network and the Watts-Strogatz (WS) model as a SW network. The nearest α_l in DT is probably caused by that the lognormal degree distribution resembles the unimodal shapes in ER and WS models rather than a power-law.



(a)



(b)

Fig. 5. Average distances and hops. Insets: the results on the exchanged paths. The dashed lines correspond the estimations of $\langle D \rangle \sim (\ln N)^{\beta_d}$ and $\langle L \rangle \sim N^{\alpha_l}$.

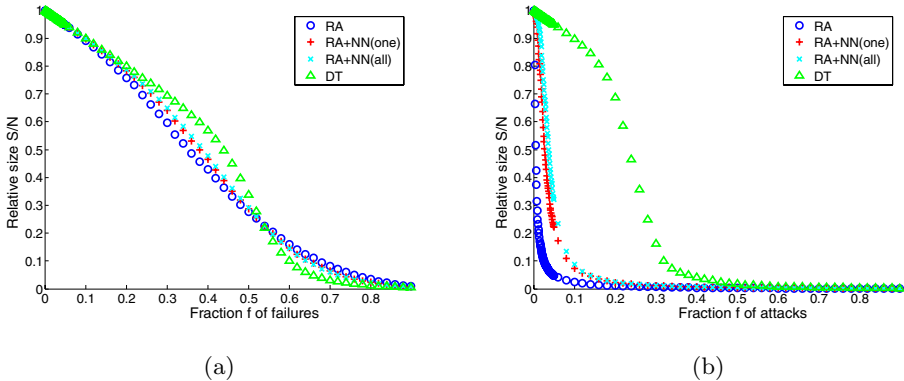


Fig. 6. Relative sizes S/N of the giant component against (a) random failures and (b) attacks on hubs with the fraction f of removed nodes. All networks have the same average degree $\langle k \rangle = 2(3N - 7)/N = 5.986$ and the minimum $k_{min} = 3$.

4 Tolerance to Failure or Attack

The fault tolerance and attack vulnerability are known as the typical SF properties [3]. We compare the tolerance of connectivity in the giant component (GC) of RAs, DTs, and RA+NN(one/all)s, when a small fraction f of the nodes is removed. Fig. 6(a) shows the similar results of the relative size S/N for the fraction of random failures, where S denotes the size of GC. Against the attacks on hubs selected in the decreasing order of degrees, Fig. 6(b) shows the improvements in RA+NN(one/all)s from the extremely vulnerable RAs caused at an early stage by the disconnection of star-like stubs concentrated at the four corners and the center nodes as shown in Fig. 2. Note that the weakly inhomogeneous DT is different from a homogeneous random network, which has the same behavior against the failure and the attack at a fraction of removed nodes [3].

5 Conclusion

We study the communication efficiency of geographical networks called RA [14],[15] and DT [18],[19] based on planar triangulation for ad hoc networks [17]. In particular, to reduce long-range links, we propose a modified model whose degree distribution follows a power-law with exponential cutoff. We find a weak disorder in the distributions of link lengths. Even in the weak disorder, we suggest the scaling laws of the shortest path length $\langle D \rangle \sim (\ln N)^{\beta_d}$ and of the minimum hop $\langle L \rangle \sim N^{\alpha_l}$ as similar to the case at the strong disorder limit [7]. From the simulation results, we conclude that RAs have a path connected by a few hops but the path length becomes long including some long-range links, while DTs have a zig-zag path connected by many hops but each link is short. Instead of the superior geometric properties [18], DTs are no longer optimal in this criteria of the minimum hops. Our model is totally balanced: the shortest

path length is the best, in addition the number of hops is the intermediate between ones in DTs and RAs. Moreover, by the structural effect, the robustness against the attacks on hubs is improved than that in RAs. Although DTs are the most robust in these network models, only it requires global configuration procedures that is unsuitable for ad hoc communication.

References

1. A.-L. Barabási, *Linked: The New Science of Networks*. Perseus (2002).
2. R.F.i Cancho, and R.V. Solé, Optimization in Complex Networks. In: R. Pastor-Satorras, M. Rubi, and A. Diaz-Guilera (eds): *Statistical Mechanics in Complex Networks*, Lecture Notes in Physics, Vol.**625**. Springer, Berlin (2003) 114-126.
3. R. Albert, and A.-L. Barabási, *Nature*, Vol.**406**, 378, (2000).
4. S.-H. Yook, H. Jeong, and A.-L. Barabási, *PNAS*, Vol.**99(21)**,13382, (2002).
5. M.T. Gastner, and M.E.J. Newman, *arXiv:cond-mat/0407680*, (2004).
6. Y. Hayashi, *IPJS Journal*, Special Issue on Network Ecology, Vol.**47(3)**, (2006) or *arXiv:physics/0512011*, (2005).
7. L.A. Braunstein, et al., *Phys. Rev. Lett.*, Vol.**91**, 168701, (2003).
8. T. Kalisky, et al., *Phys. Rev. E*, Vol.**72**, 025102, (2005).
9. B. Tadic, and S. Thurner, *Physica A*, Vol.**332**, 566, (2004) and **346**, 183, (2005).
10. S.S. Manna, and S. Parongama, *Phys. Rev. E*, Vol.**66**, 066114, (2002).
11. R. Xulvi-Brunet, and I.M. Sokolov, *Phys. Rev. E*, Vol.**66**, 026118, (2002).
12. C.P. Warren, L.M. Sander, and I.M. Sokolov, *Phys. Rev. E*, Vol.**66**, 056105, (2002).
13. D. ben-Avraham, et al., *Physica A*, Vol.**330**, 107, (2003).
14. T. Zhou, G. Yan, and B.-H. Wang, *Phys. Rev. E*, Vol.**71**, 046141, (2005).
15. J.P.K. Doye, and C.P. Massen, *Phys. Rev. E*, Vol.**71**, 016128, (2004).
16. J.S. Andrade,Jr., et al., *Phys. Rev. Lett.*, Vol.**94**, 018702, (2005).
17. P. Bose, and P. Morin, *SIAM J. of Computing*, Vol.**33(4)**, 937, (2004).
18. K. Imai, *IEICE Trans. on Infor. and Syst.*, Vol.**83-D(3)**, 428, (2000).
19. A. Okabe, et al., *Spatial Tessellations*, 2nd ed. John Wiley (2000).
20. J.M. Keil, and C.A. Gutwin, *Discrete Compt. Geom.*, Vol.**7**, 13, (1992)
21. L.A.N. Amaral, et al., *PNAS*, Vol.**97(21)**, 11149, (2000).
22. R Cohen et al., Structural properties of scale-free networks. In: S. Bornholdts, and H.G. Shuster (eds): *Handbook of Graphs and Networks*. Wiely-VCH, New York (2002) 85-110.

Collaborative Tagging as a Tripartite Network

Renaud Lambiotte and Marcel Ausloos

SUPRATECS, Université de Liège, B5 Sart-Tilman, B-4000 Liège, Belgium
{Renaud.Lambiotte, Marcel.Ausloos}@ulg.ac.be

Abstract. We describe online collaborative communities by tripartite networks, the nodes being persons, items and tags. We introduce projection methods in order to uncover the structures of the networks, i.e. communities of users, genre families... The structuring of the network is visualised by using a tree representation. The notion of diversity in the system is also discussed.

1 Introduction

Recently, new kinds of websites have been dedicated to the sharing of people's habits and tastes, examples including their preferences in music, scientific articles, movies, websites... These sites allow members to upload from their own computer a library that characterises their habits in the corresponding topic (an iTunes music library for instance), and next to create a web page containing this list of items. Additionally, the website proposes the users to discover new content by comparing their taste with that of other users, thereby helping them discover new musics/books/websites... that should (statistically) fit their profile.

This method rests on a feedback between the users and a central server, and is usually called collaborative filtering. The emergence of these collaborative websites answers the needs of Internet users to retrieve useful and coherent informations from the millions of pages and data that form the Web. The main particularities of collaborative systems are: (i) their non-commercial purpose, even though the frontier with commercial companies is more and more vague (see for instance the acquisition of *del.icio.us* by *Yahoo* in November 2005); (ii) their transparency, namely these sites are relatively open and do not hide the profiles of each user, contrary to Amazon for instance. From a scientific point of view, this transparency opens perspectives in order to perform large scale experiences (including thousands of people) on taste formation, quantitative sociology, musicology... The available data also suggest alternative methods in order to perform large scale classifications of music/science/internet. Those subdivisions should be based on the intrinsic structure of the audience of the items.

In parallel with this sharing and statistical comparing of content, collaborative websites usually propose tagging possibilities. This process, called "folksonomy" (short for "folk taxonomy") means that the websites allow users to publicly tag their shared content, the key point being that their tag is not only accessible to themselves, but also to the whole ensemble of users. For instance, in the case of music sharing habits, a group like *The Beatles* is described in different ways,

i.e. *pop*, *60s*, *britpop*..., that depend on the different backgrounds, tastes, music knowledge or *network of acquaintances*... of the users.

2 Methodology

The structure of collaborative websites can be viewed as a tripartite network. Namely, it is a network composed of three kinds of nodes: i) the persons or users μ ; ii) the items i that can be music groups or scientific articles; iii) the tags I that are used by the person μ to describe the item i . Depending on the systems under consideration, a person can use one or several tags on each item. The resulting network can be represented by a graph where edges run between the item i and the user μ , passing through the tag I . Moreover, a weight is attributed to each link depending on the number of tags given by μ to i . For instance, if μ uses two tags for i , the weight of the links is $\frac{1}{2}$.

Let us note n_U the number of users, n_{It} the number of items, and n_T the number of tags in the considered sample. Consequently, each listener μ can be characterised by the $n_{It} \times n_T$ matrix $\overline{\overline{\sigma}}^\mu$:

$$\overline{\overline{\sigma}}^\mu = \begin{pmatrix} 0 & \dots & 1/2 & \dots & 1/2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 1/3 & \dots & 1/3 & \dots & \dots & 1/3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (1)$$

where $\overline{\overline{\sigma}}_{iI}^\mu$ denotes the weight of tag I in its description of i , so that $\sum_I \overline{\overline{\sigma}}_{iI}^\mu = 1$ if μ owns i and zero otherwise. Each item and each tag is also characterised by similar matrices that we note $\overline{\overline{\gamma}}^i$ and $\overline{\overline{\alpha}}^I$ respectively.

A common way to simplify the analysis of multi-partite networks consists in projecting them on lower order networks, i.e. unipartite or bipartite networks. In the following, we only focus on the correlations between two kinds of nodes, for instance between the users and the items. To do so, we first reduce the tripartite network to a bipartite one by summing over all nodes of one kind, thereby neglecting possible correlations between the three kinds of nodes. For instance, the bipartite network users-item is obtained by summing over all tags, so that each listener μ is now described by the n_{It} -vector $\overline{\overline{\sigma}}_I^\mu$:

$$\overline{\overline{\sigma}}_I^\mu = (\dots, 1, \dots, 0, \dots, 1, \dots), \quad (2)$$

the index running over all items, and where $\overline{\overline{\sigma}}_I^\mu = \sum_I \overline{\overline{\sigma}}_{iI}^\mu$. The items are characterised by the n_U -vector $\overline{\overline{\gamma}}_I^i = (\dots, 1, \dots, 0, \dots, 1, \dots)$. These vectors are signatures of the users/items, that account for their interests/audience. In the case of music, we call these vectors the *music signatures* of people and groups.

In order to project the bipartite network on a unipartite one, we look at the correlations between two nodes of the same kind, relatively to his behaviour with another kind. For instance, one may look how persons μ and λ are correlated by using common items. To do so, we introduce the symmetric correlation measure:

$$C_{CF}^{\mu\lambda} = \frac{\bar{\sigma}_{|I}^{\mu} \cdot \bar{\sigma}_{|I}^{\lambda}}{|\bar{\sigma}_{|I}^{\mu}| |\bar{\sigma}_{|I}^{\lambda}|} \equiv \cos \theta_{\mu\lambda} \tag{3}$$

where $\bar{\sigma}_{|I}^{\mu} \cdot \bar{\sigma}_{|I}^{\lambda}$ denotes the scalar product between the two n_{I_t} -vector, and $||$ its associated norm. This correlation measure, that corresponds to the cosine of the two vectors in the n_{I_t} -dimensional space, vanishes when the persons have no common item, and is equal to 1 when their item libraries are strictly identical.

At this level, the search for structures requires the analysis of large correlation matrices, and the uncovering of connected blocks that could be identified as families/genres/communities. In order to extract families of alike elements from the correlation matrix \mathbf{C} , we define the filter coefficient $\phi \in [0, 1[$ and filter the matrix elements so that $C_{\phi}^{ij} = 1$ if $C^{ij} > \phi$ and $C_{\phi}^{ij} = 0$ otherwise. Starting from $\phi = 0.0$, namely a fully connected network, increasing values of the filtering coefficient remove less correlated links and lead to the shaping of well-defined islands, completely disconnected from the main island.

A branching representation of the community structuring [1] is used to visualise the process. To do so, we start the procedure with the lowest value of $\phi = 0.0$, and we represent each isolated island by a square whose surface is proportional to its number of internal elements. Then, we increase slightly the value of ϕ , e.g. by 0.05, and we repeat the procedure. From one step to the next step, we draw a bond between emerging sub-islands and their parent island. The filter is increased until all bonds between nodes are eroded (that is, there is only one node left in each island). Let us note that islands composed of only one element are not depicted for the sake of clarity. Applied to the above correlation matrix C^{ij} , the tree structure gives some insight into the specialisation by following branches from their source toward their extremity.

By construction, the above procedure unambiguously attributes to each element a hierarchical set of categories [2]. Consequently, starting from collaborative filtering that is a non-exclusive and non-hierarchical process, we have arrived to an exclusive and hierarchical structure that may be viewed as a taxonomy. This relation could have helpful applications in order to automatically structure content in systems without a central authority.

3 Applications: Measuring Diversity

Amongst others, this work provides tools in order to compare the tastes and interests of different persons, as well as to measure their *diversity*. Practically, let us focus on music collaborative websites and consider the case of two users μ_1 and μ_2 who own a list of music groups, each of them characterised by a spectrum of genres. From this knowledge, one would like to find a quantitative measure of the diversity of the persons, and a way to measure whether they have a similar taste. Let us note τ_{μ_1} and τ_{μ_2} the vector of genres characterising μ_1 and μ_2 , where

$$\tau_{\mu_1} = (\tau_{\mu_1 1}, \dots, \tau_{\mu_1 I}, \dots, \tau_{\mu_1 n_T}). \tag{4}$$

$\tau_{\mu_1 I}$ is the number of times that the tag I is associated to an item of μ_1 and n_T is the total number of tags in the system. A naive way to study diversity consists in implicitly assuming that all tags have different meaning and in characterising a person by the width of the distribution of τ . This is what we have done in ref.[3], where we defined a probabilistic entropy in order to measure these fluctuations. It is nonetheless an oversimplification that does not take into account the correlations between the tags, i.e. the fact that tags may have more or less equivalent meanings.

A more refine measure of diversity should require a proper counting of the *categories* to which the user belongs. To do so, we propose to visualise the branches and sub-branches of the hierarchical tree in which the user is more active than the average. Let us assume that, at some level of the filtering, an island (the node of one branch in the tree representation) is composed of K tags, say $I_1, \dots, I_i, \dots, I_K$. Let us denote $\tau_{I_i}^S$ the total number of times the tag I_i is used in the sample, while, as defined above, $\tau_{\mu I_i}$ is the total number of times I_i is tagged to the items belonging to μ . The above island, composed of K genres, is then characterised by:

- $p^S = (\sum_{i=1}^K \tau_{I_i}^S) / (\sum_{I=1}^{n_T} \tau_I^S)$, that gives the empirical probability that a tag used in the sample belongs to the considered island.

- $p^\mu = (\sum_{i=1}^K \tau_{I_i}^\mu) / (\sum_{I=1}^{n_T} \tau_I^\mu)$, that is the probability that a tag used on an item of μ belongs to the same island.

The activity of the user in the island is simply evaluated by looking at the ratio $r = p^\mu / p^S$. By construction, this quantity is bigger than 1 if the user owns many groups belonging to this island, and smaller than 1 otherwise.

Applying the method to all the nodes of the tree representation, and using a colour representation in order to represent the value of r , i.e. the nodes are printed in a colour ranging from green (low r) to blue (high r) [4], the user's diversity may be visualized. Moreover, different users may be compared by looking whether they are active in the same branches or in different branches.

During the poster presentation, the above techniques will be applied to empirical data extracted from websites specialised in music, e.g. *audioscrobbler.com* and *musicmobs.com*, and in scientific articles, i.e. *citeulike.com*.

Acknowledgments. This work has been supported by European Commission Project CREEN FP6-2003-NEST-Path-012864.

References

1. R. Lambiotte and M. Ausloos, *Phys. Rev. E*, **72**, 066107 (2005)
2. S. Golder and B. A. Huberman, cs.DL/0508082
3. R. Lambiotte and M. Ausloos, *EPJB*, in press; physics/0509134
4. R. Lambiotte and M. Ausloos, cs.DS/0512090

Author Index

- Aalto, Samuli IV-420
Abawajy, J.H. IV-1015, IV-1071
Abbas, Ghulam I-802
Abbaspour, Amin III-578
Abbod, M.F. I-993
Abed, E.H. III-448
Abramson, David I-720
Absil, P.-A. I-210
Agarwal, Pankaj K. III-409
Aguado González, S. II-350
Aguilar-Saborit, Josep I-156
Ahmad, Jamil II-887
Ahn, Chunsoo I-952
Ahn, Hye-Young IV-894
Ahn, Hyun Gi I-989
Akbal, Ayhan IV-631, IV-638
Akbar, Ali Hammad II-1073
Akelik, Volkan III-481
Akdin, B. I-372
Akioka, Sayaka I-242
Akritas, Alkiviadis II-486
Aksyonov, Konstantin A. III-879
Aldana-Montes, Jose F. III-936
Alemani, Davide II-70
Alexandrov, Vassil I-868, II-595, II-603,
III-632, III-640
Alfonsi, Giancarlo I-465
Alípio, Pedro III-240
Al-khalifah, Ali I-868
Almeida, Francisco I-872
Alonso, J. III-313
Alonso, Pedro I-348
Altınakar, Mustafa Siddik II-58
Altintas, İlkay III-69, III-920
Alves, Domingos I-1005, III-297
Amghar, T. I-603
Amorim, Ronan M. I-68
Anagnostou, Miltiades I-579, I-892
Anai, Hirokazu II-462
Ang, Chee-Wei IV-260
Angelova, Donka III-624
Anguita, M. II-518
Antaki, James IV-855
Aramudhan, M. IV-388
Argyarakis, Panos III-1048
Arora, Nitin I-16
Arrowsmith, J. Ramon III-920
Arslan, Ahmet II-247
Artoli, Abdel Monim II-78
Assous, Franck IV-623
Asthana, A. III-161
Atanassov, Emanouil III-616
Ausloos, Marcel III-1114
Avci, Mutlu IV-615
Avila, Andres I-856
Awan, Asad III-465
Babik, Marian III-980
Babuška, I. III-530
Bachman, Timothy IV-855
Baddeley, B. II-871
Bader, M. I-673
Bae, Jungsook II-1033
Bae, Sung Eun I-595
Baek, Myung-Sun II-969, II-1058
Bagheri, Ebrahim I-588
Bajaj, C. III-530
Bajuelos, António Leslie II-255
Baker, C.G. I-210
Baker, Mark II-953
Balas, Lale I-814
Balcan, Duygu III-1083
Baldrige, Kim III-69
Balicki, Jerzy III-863
Balik, Hasan H. IV-631, IV-638
Baliś, Bartosz II-542
Balla, Sudha II-822
Balogh, Zoltan III-980
Balos, Kazimierz IV-1039
Banaś, K. III-743
Bandini, Stefania III-289
Banicescu, Ioana II-430
Bao, Yukun I-728, IV-308, IV-517
Barabási, Albert-László III-417
Barbosa, Ciro B. I-68
Baronikhina, Anastasia A. III-879
Barrientos, Ricardo J. I-611
Barsky, Prof. Brian A. II-215

- Bartlett, Roscoe A. IV-525
 Bartocci, Ezio III-1012
 Baru, Chaitan III-920
 Bass, J. III-530
 Battiato, Sebastiano II-334
 Baumgartner, Gerald I-267
 Bauschlicher Jr., C.W. III-128
 Beagley, N. II-871
 Becker, Michael F. I-443
 Beezley, Jonathan D. III-522
 Bell, Ron I-7
 Belmonte Fernández, Ó. II-350
 Benavent, Xaro III-13
 Bergamaschi, L. IV-685
 Bernabeu, Miguel O. I-348
 Bernard, Julien IV-999
 Bernhard, Fabrice IV-236
 Bernholdt, David E. I-267
 Bernier, J.L. II-518
 Bernot, Gilles II-887
 Bernreuther, Martin II-161
 Bernstein, D.S. III-489
 Bertozzi, Luigi IV-831
 Berzins, M. II-147
 Bhana, Ismail II-581
 Bhattacharjee, Apurba K. I-387
 Bianchini, Germán I-539
 Bie, Rongfang II-775, IV-781
 Biely, Christoly III-1067
 Bieniasz, Sławomir III-759
 Biros, George III-481
 Biscay, R.J. I-132
 Bishop, Marvin III-608
 Blais, J.A.R. III-48
 Blelloch, Guy E. II-799
 Blower, J.D. III-996
 Bonizzoni, Paola II-622
 Borowski, Stefan I-196
 Botana, Francisco II-470
 Bourchtein, Andrei I-258
 Bourchtein, Ludmila I-258
 Bourgeois, Anu II-678
 Branford, Simon III-632, III-640
 Brendel, Ronny II-526
 Breuer, Peter T. IV-765
 Brill, Downey III-401
 Brinza, Dumitru II-767
 Briseid, Sverre IV-204
 Brogan, David III-570
 Brown, Martin IV-773
 Browne, J.C. III-530
 Brunst, Holger II-526
 Bu, Jiajun I-449
 Bubak, Marian II-542
 Buendia, Patricia II-807
 Buffle, Jacques II-70
 Bunde, Armin III-1048
 Bungartz, Hans-Joachim II-161
 Burguillo-Rial, J.C. III-815
 Bustos, Benjamin IV-196
 Byun, Yanga I-276, I-284
 Caballero-Gil, P. III-337
 Cai, Guoyin I-292, I-876, III-1
 Cai, Yang I-1001, IV-870
 Caliari, M. IV-685
 Calle, Eusebi IV-136
 Camahort, Emilio II-287, II-310
 Campos, Celso II-310
 Campos, Fernando Otaviano I-68, I-76
 Cannarozzi, Gina M. II-630
 Cao, Chunxiang III-9
 Cao, Jian III-948
 Cao, Wuchun III-9
 Cao, Yuanda IV-81
 Čapkovič, František III-176
 Cappello, Angelo IV-831
 Carbonell, F. I-132
 Cariño, Ricolindo L. II-430
 Carnahan, Joseph III-570
 Caron, David III-514
 Carpenter, Bryan II-953
 Carr, Nathan A. IV-228
 Carvalho, Luis Alfredo V. de I-842
 Carvalho, Paulo III-240
 Castelló, Pascual II-263
 Čepulkauskas, Algimantas II-407
 Cetnarowicz, Krzysztof III-775, III-823,
 III-855
 Chae, Kijoon II-1024
 Chai, Zhilei I-1043
 Chakravarty, Manuel M.T. II-920
 Chambarel, André II-50
 Chandrasekar, J. III-489
 Chang, Chun-Hyon IV-280
 Chang, F.K. III-456
 Chang, Hao-Li IV-878
 Chang, Kungyen I-226
 Chang, Kyungbae IV-987
 Chantzara, Maria I-579

- Charoy, François III-976
 Chatterjee, Abhijit III-77
 Chaturvedi, Alok III-433
 Chauve, Cedric II-783
 Chen, Chun I-449
 Chen, Guihai IV-404
 Chen, Jianming IV-501
 Chen, Juan II-646, II-904
 Chen, Junliang IV-104
 Chen, Lei IV-938
 Chen, Ling II-646
 Chen, Ming-Jen IV-184
 Chen, Shudong III-1004
 Chen, Su-Shing II-830
 Chen, Yangzhou II-478
 Chen, Yibing I-851
 Chen, Yixin II-646
 Chen, Yongqiang I-896
 Chen, Yu-Sheng I-1026
 Chen, Zhanglong I-1043
 Chen, Zhengxin IV-476, IV-485
 Cheng, Guang IV-144
 Cheng, Haiying I-1047
 Cheng, Jingde IV-797
 Cheng, Junbo I-851
 Cheng, Junxia I-851
 Cheng, Ruixing IV-87
 Cheng, Shiduan IV-128
 Cheng, Wang-Cho IV-260
 Chi, Hongmei IV-773
 Chiang, Tzu-Chiang II-1008
 Chinnasarn, Krisana I-403
 Chinnasarn, Sirima I-403
 Cho, Choongho IV-168
 Cho, Dong-Jun II-1058
 Cho, Han Wook IV-244
 Cho, Insook III-1040
 Cho, Jin-Woong II-1041
 Cho, Keumwon III-972
 Cho, KumWon IV-293
 Cho, Kyu Bong II-587
 Cho, Sung-Jin I-1067
 Cho, Yongyun I-965, II-510
 Cho, Yookun IV-946
 Choi, Bumghi I-63
 Choi, Hyoung-Kee II-961
 Choi, Jaeyoung I-965, I-1059, III-972
 Choi, Jeong-Yong IV-25
 Choi, Jin-Ghoo IV-160
 Choi, Jin-Hee IV-160, IV-172
 Choi, Kee-Hyun III-895, III-899
 Choi, Min-Hyung I-308, I-490
 Choi, Seung-Hyuk I-969
 Choi, Un-Sook I-1067
 Chong, Kiwon IV-902
 Choo, Hyunseung I-948, I-960, I-989,
 II-1089
 Chopard, Bastien II-70, IV-653
 Choppella, Venkatesh I-267
 Chou, Chien-Lung I-900
 Chover, Miguel II-263
 Chow, Peter II-34
 Chowdhury, A.K. III-161
 Christov, N.D. I-697
 Chu, Yuan-Sun IV-184
 Chung, Min Young I-969, I-989
 Chunguo, Wu I-547
 Chuyi, Song I-547
 Ciarlet Jr., Patrick IV-623
 Cięciwa, Renata III-823
 Cirak, Fehmi II-122
 Ciszewski, Stanisław III-759
 Čivilis, Alminas I-1034
 Ciuffo, Leandro N. I-68
 Clark, James S. III-409
 Clarke, Bill I-218
 Cocu, Adina I-172
 Coen, Janice III-522
 Cohen, Reuven III-1048
 Cokuslu, Deniz I-571
 Cole, Martin J. III-393
 Cole, Murray II-929
 Collier, R. III-727
 Colling, D.J. III-956
 Collins, Timothy M. II-807
 Combes, P. IV-653
 Comet, Jean-Paul II-887
 Conner, Jeffery III-920
 Cornford, Dan III-586
 Corradini, Flavio III-1012
 Cortés, Ana I-539
 Costa-Montenegro, E. III-815
 Cotoi, I. II-26
 Cox, Simon J. III-928
 Crosby, Christopher J. III-920
 Crăciun, Marian Viorel I-172
 Cuadrado, J.J. IV-789
 Cui, Pingyuan II-478
 Culebras, R. I-395
 Curley, Martin I-4

- Cycon, Hans L. II-1050
 Czarnul, Pawel III-944
 Czekierda, Lukasz III-940
- Dagdeviren, Orhan I-571
 da Silva, Fabrício A.B. I-1005, III-297
 Dai, Wenchao I-1047
 Dai, Yafei IV-412, IV-428
 Dailyudenko, Victor F. I-846
 Dal Negro, Marco III-264
 Danelutto, M. II-937
 Danilecki, Arkadiusz I-753
 Darema, Frederica III-375
 Darlington, John III-964
 Das, Abhimanyu III-514
 Das, A.K. III-161
 Dauvergne, Benjamin IV-566
 Davila, Jaime II-822
 Dazzi, P. II-937
 Degond, Pierre II-1
 Deiterding, Ralf II-122
 De la Cruz, H. I-132
 de la Encina, Alberto II-207
 Delaplace, Franck III-1056
 Della Vedova, Gianluca II-622
 Del Vecchio, David I-681
 De Paoli, Serge IV-999
 Demkowicz, L. III-530
 Deng, Hui IV-17
 Deussen, Oliver IV-196
 Deville, Michel II-58
 Dhamdhere, Kedar II-799
 Dhariwal, Amit III-514
 Díaz, Manuel II-912
 Dickens, L.W. III-956
 Dieci, Luca IV-677
 Dieter, Bill I-226
 Dietz, Hank I-226
 Dikshit, Anupam II-830
 Diller, K.R. III-530
 Dimov, Ivan III-632, III-640
 Ding, Koubao I-482
 Ding, Shifei I-777
 Ding, Wei IV-112, IV-120, IV-144
 Dobnikar, Andrej III-345
 Dogan, Kaan II-996
 Dokken, Tor IV-204
 Dondi, Riccardo II-622
 Donetti, Luca III-1075
 Dong, Hongbin III-216
- Dong, Jin-xiang IV-839
 Dong, Yabo IV-57
 Dornaika, Fadi I-563
 Dou, Wen-hua I-1030
 Douglas, Craig C. III-393, III-522
 Draganescu, Andrei III-481
 Drezewski, Rafał III-871, III-908
 Drummond, Arielle IV-855
 Duan, X. I-372
 Dumitriu, Luminița I-172, II-199
 Dyshlovenko, Pavel IV-599
- Easterday, Scott IV-582
 Efendiev, Yalchin III-393
 Eleftheriou, Maria II-846
 Ellis, Carla III-409
 El Yacoubi, Samira III-360
 Engelmann, Christian II-573
 Ensan, Faezeh I-588
 Eom, Young Ik IV-356
 Erciyes, Kayhan I-571
 Erlebacher, Gordon II-177
 Erzan, Ayse III-1083
 Escrivá, Miguel II-287
 Evans, Deidre W. IV-773
 Ewing, Richard III-393
- Falzon, Chantal T. III-82
 Fan, Liangzhong II-367
 Fan, Zhiping III-601
 Fang, Liqun III-9
 Fang, Wei III-847
 Fangohr, Hans II-139
 Fantozzi, Silvia IV-831
 Farhat, C. III-456
 Farinella, Giovanni Maria II-334
 Farinelli, Simone IV-324
 Fathy, M. I-744
 Feixas, Miquel II-263
 Felici, G. IV-460
 Feller, Scott II-846
 Feng, Dan I-1063, III-671, IV-396
 Feng, Gang IV-645
 Feng, Y. III-530
 Feng, Yi IV-468
 Fernández, A. III-313
 Fernández de Vega, F. III-281
 Fernández, J. II-518
 Fernando, Terrence III-60
 Ferrari, T. III-956

- Fertin, Guillaume II-622, II-783
 Fesehayé Kassa, Debessay IV-65
 Fidanova, Stefka I-1009
 Figueiredo, Renato III-546
 Filatyev, Sergei III-433
 Fisher, Randy I-226
 Fitch, Blake G. II-846
 Fjeldly, T.A. IV-607
 Fladmark, Gunnar II-102
 Flikkema, Paul G. III-409
 Forestiero, Agostino IV-1047
 Fort, H. III-313
 Fortes, José III-546
 Forth, Shaun A. IV-558
 Fougère, Dominique II-50
 Freundl, C. II-185
 Fritz, Nicolas IV-200
 Froelich, Wojciech III-839
 Fu, Chong I-826
 Fu, Karl I-1001
 Fu, Qing-hua IV-878
 Fu, Xuezheng II-678
 Fuentes, D. III-530
 Fujimoto, Richard II-41, III-425
 Funika, Włodzimierz II-534, II-549
 Furlan, Luciana B. I-1005
 Furlinger, Karl II-494
 Fúster-Sabater, A. III-337
- Gaaloul, Khaled III-976
 Gagliardi, Henrique F. I-1005, III-297
 Gaitán, Rafa II-287
 Galante, M.A. IV-460
 Galceran, Josep II-70
 Gallivan, K.A. I-210
 Gallos, Lazaros K. III-1048
 Gálvez, A. II-414
 Gamalielsson, Jonas II-879
 Gan, Honghua I-204
 Ganzha, Maria III-208
 Gao, Lijun IV-501
 Gao, Wenzhong II-430
 Gao, Xiaodong III-601
 Gao, Xiaoyang I-267
 Gao, Zhigang IV-918
 García, Víctor M. I-324
 Garšva, Gintautas IV-364
 Garzón, E.M. II-106
 Gashkov, Igor I-912
 Gasparo, Maria Grazia IV-677
- Gastineau, Mickaël II-446
 Gavrilenko, Vladimir I. III-89
 Gay, David M. IV-525
 Geist, Al II-573
 Gelfand, Alan III-409
 George, E. Olusegun II-694
 Germain, Robert S. II-846
 Gerndt, Michael II-494
 Ghattas, Omar III-481
 Giampapa, Mark II-846
 Giannoutakis, Konstantinos M. I-506
 Giering, Ralf IV-591
 Gill, Ofer H. II-638, II-654
 Gilmore, Stephen II-929
 Girdzijauskas, Stasys IV-364
 Glut, Barbara II-302
 Godart, Claude III-976
 Golubchik, Leana III-514
 Gomes, André Severo Pereira III-97
 Goncharova, Natalia V. III-879
 Gong, Jian I-1022, IV-112, IV-120,
 IV-144
 Gonnet, Gaston H. II-630
 González, Daniel I-872
 González, J. I-649
 González, Luis III-305
 González-Castaño, F.J. III-815
 González de-la-Rosa, Juan-José I-316
 Gopalan, B. II-871
 Gopinath, K. III-679
 Gore, Jay III-433
 Górriz, J.M. I-234, I-316, I-356,
 I-395, I-649
 Goscinski, A.M. IV-1015
 Goulard, Frédéric I-332
 Govaerts, W. II-391
 Govindan, Ramesh III-514
 Grabska, Ewa III-883
 Graham, Richard L. II-945
 Grama, Ananth III-465
 Gravvanis, George A. I-506
 Grivel, E. I-697
 Grochowski, M. III-783
 Grossfield, Alan II-846
 Grunewaldt, Lars II-565
 Grzech, Adam III-224
 Grzesiak-Kopeć, Katarzyna III-883
 Guan, Ximeng I-250
 Guensler, R. III-425
 Guibas, L.J. III-456

- Guisado, J.L. III-281
 Guitart, Jordi I-84
 Guodong, Yuan I-864
 Guo, Jianping I-292, I-876, III-1, III-9
 Guo, Wu II-223
 Gurel, Guray II-996
 Gurov, Todor III-616
 Gusfield, Dan II-618
 Gutsev, G.L. III-128
 Guzy, Krzysztof II-542
- Habala, Ondrej III-980
 Haffegée, Adrian II-595, II-603
 Hagen, Trond Runar IV-204, IV-220
 Hager, Svenja IV-340
 Haines, K. III-996
 Hajiaghayi, M.T. II-758
 Halperin, Eran II-799
 Ham, Eunmi IV-894
 Han, Chang-Wook IV-862
 Han, Joohyun I-965
 Han, JungHyun III-40
 Han, Kijun I-940, IV-180
 Han, Kyungsook I-276, I-284
 Han, SeungJae II-1101
 Han, Sunyoung I-936, II-1081, IV-260
 Han, Wook-Shin III-648
 Handzlik, Piotr II-549
 Harman, Mark IV-740
 Harris, J. Clay III-393
 Harrison, A.B. III-996
 Harrison, Ken III-401
 Harrison, Robert II-710
 Harrison, Rodrigo III-1091
 Hartell, Mark G. I-387
 Hartono, Albert I-267
 Hascoët, Laurent IV-566
 Hassoun, Y. III-956
 Havlin, Shlomo III-1048
 Hayashi, Yukio III-1106
 Hazle, J. III-530
 He, Gaiyun I-822
 He, Jieyue II-710
 He, Jing II-1069, IV-509
 He, Jingwu II-750
 He, Yulan II-718
 Hegeman, Kyle IV-228
 Heo, Junyoung IV-946
 Hermer-Vazquez, Linda III-546
 Hernandez, Gonzalo I-856, III-1091
- Hernández Encinas, L. II-438
 Hicks, Rickey P. I-387
 Hidalgo-Herrero, Mercedes II-207
 Hill, Judith III-481
 Hiller, Stefan IV-196
 Hincapie, Doracelly I-920
 Hiroaki, Deguchi II-490
 Hirokazu, Hashiba II-490
 Hluchy, Ladislav III-980
 Honavar, Vasant G. III-440
 Hong, Choong Seon II-1016
 Hong, Jiman IV-946, IV-970, IV-991
 Hong, Kwang-Seok IV-886
 Hong, Min I-308, I-490
 Hong, Tzung-Pei I-1026
 Horie, Daisuke IV-797
 Horng, Gwoboa I-1026
 Hovland, Paul IV-574, IV-582
 Howard, I.C. I-993
 Hrach, Rudolf I-806
 Hsiao, Chieh-Ming IV-757
 Hsu, Chung-Chian IV-757
 Hsu, Ji-Hsen II-295
 Hu, Dewen I-689
 Hu, Hongyu IV-81
 Hu, Xiaoping I-689
 Hu, Xiaoyong I-838
 Hu, Yincui I-292, I-876, III-1
 Huang, Changqin I-838
 Huang, Guangyan II-1069
 Huang, Houkuan III-216
 Huang, Jin I-1051
 Huang, Wayne IV-188
 Huang, Wei I-728, IV-308, IV-493,
 IV-517
 Huang, Xianglin I-997
 Huang, Xin I-411
 Huang, Y. III-554
 Huang, Yan-Ping IV-757
 Huang, Yueh-Min II-1008
 Huerta, Joaquin II-310
 Huh, Eui-Nam I-960
 Humphrey, Marty I-681
 Hunke, Elizabeth C. IV-533
 Hunter, M. III-425
 Huo, Mingxu I-482
 Hurtado, Pablo I. III-1075
 Hwang, Ho Seon I-944
 Hwang, Ho-Yon IV-264
 Hwang, In-Yong I-1018

- Hwang, Jae-Hyun IV-172
 Hwang, Tae Jin I-944
 Hwang, Yoon-Hee I-1067
- Iavernaro, Felice IV-724
 Iglesias, A. II-383, II-414
 Im, Sungbin II-992
 Imielinska, Celina IV-822
 Iniguez, B. IV-607
 Ipanaque, R. II-383
 Irwin, Mary Jane I-242
 Iskandarani, Mohamed III-393
 Iskra, K.A. III-281
 Ismail, Suhaila IV-1071
 Issakova, Marina I-928
- Jackson, Steven Glenn II-422
 Jaeger-Frank, Efrat III-920
 Jain, K. II-758
 Jakop, Yanto I-522
 Jakubowska, Malgorzata I-498
 Jaluria, Y. III-473
 Jamieson, Ronan II-595, II-603
 Janicki, Aleksander IV-301
 Janik, Arkadiusz IV-1023
 Janik, Pawel I-300
 Jankowski, Robert III-56
 Jarzab, Marcin IV-1039
 Javaheri Javid, Mohammad Ali III-367
 Jedrzejowicz, Piotr III-719
 Jeon, Jun-Cheol III-329, IV-661
 Jeon, Segil IV-272, IV-293
 Jeong, Karpjoo IV-264, IV-293
 Jeong, Taek Sang III-40
 Jeong, Taikyeong T. I-443, I-761,
 III-105, III-113
 Jeong, Yoon-Seok IV-280
 Jeong, Yoon-Su I-908
 Jeras, Iztok III-345
 Jermann, Christophe I-332
 Jhang, Seong Tae IV-979
 Jhon, Chu Shik IV-979
 Jia, Jinyuan II-342
 Jia, Zupeng I-851
 Jiang, Gangyi II-367
 Jiang, Hong IV-396
 Jiang, Peng I-794, IV-693
 Jie, Min Seok I-108
 Jimenez, J.C. I-132
 Jiménez-Morales, F. III-281
- Jin, Guang IV-57
 Jin, Hai I-1051, IV-380, IV-1055
 Jin, Shiyao I-769
 Jin, Xiaogang I-140, III-1032
 Jin, Xin II-775, IV-781
 Jin, Yu Xuan IV-264
 Jin, Yuehui IV-128
 Jingqing, Jiang I-547
 Johnson, Chris R. I-3, II-147, III-393
 Johnson, David II-581
 Johnson, John IV-188
 Joneja, Ajay II-342
 Jones, Anthony III-586
 Jones, Edward L. IV-773
 Joo, Bok-Gyu IV-260
 José L. Valcarce II-470
 Juhasz, Zoltan I-830
 Julià, Carme I-555
 June, Sun-Do II-1041
 Jung, Bokrae IV-152
 Jung, Eui Bong IV-244
 Jung, Jason J. III-244
 Jung, Myoung-Hee I-969
 Jung, Seunho IV-293
 Jung, Ssang-Bong II-1041
 Jung, Sungwon II-1065
 Jurczyk, Tomasz II-302
 Juszczyzyn, Krzysztof III-224
- Kaczmarek, Pawel L. I-904
 Kaihuai, Qin I-864
 Kaiser, Tim I-379
 Kamci, A. Kerim II-996
 Kaminski, Thomas IV-591
 Kaminski, Wieslaw A. II-94
 Kanaujia, A. III-554
 Kang, Byung-Heon III-329
 Kang, Byung-Su II-969, II-1058
 Kang, Dazhou IV-95
 Kang, Eui Chul II-371
 Kang, Lishan I-340
 Kang, Mikyung IV-962, IV-970
 Kang, Minho IV-152
 Kang, Minhyung II-977
 Kar, T. I-372
 Karaivanova, Aneta III-616
 Karakaya, Ziya II-375
 Karam, Noha Makhoul I-148
 Karcanias, Nicos I-798, II-399
 Karimabadi, Homa II-41

- Karl, Wolfgang II-502
 Karniadakis, G.E. III-538
 Karpowicz, Michał III-791
 Kasperska, Elżbieta I-24
 Kasprzak, Andrzej I-100
 Katarzyniak, Radosław III-224, III-891
 Kawasaki, Yohei IV-436
 Keim, Daniel IV-196
 Keller, Gabriele II-920
 Kempe, David III-514
 Kennedy, Catriona III-562
 Keriven, Renaud IV-212, IV-236
 Kernan, Warnick I-179
 Kharche, Rahul V. IV-558
 Khattri, Sanjay Kumar I-860, II-102,
 II-239
 Khonsari, A. I-744
 Khrenov, Alexey A. III-879
 Khrustaleva, Ekaterina Yu. I-117
 Kim, ByungChul II-1033
 Kim, Byunggi III-1040
 Kim, Chang-Hun II-279
 Kim, Choelmin IV-962
 Kim, Dae Sun II-1016
 Kim, Daehee IV-922
 Kim, Dongryung III-807
 Kim, Duck Bong II-371
 Kim, Han-Doo I-1067
 Kim, Hanil IV-962
 Kim, H.-K. III-425
 Kim, Hwa-sung IV-954
 Kim, Hyeoncheol II-830
 Kim, HyoJin II-1101
 Kim, Hyung-Jun III-895, III-899
 Kim, Hyunmyung IV-293
 Kim, Hyunsook I-940
 Kim, Hyun-Sung I-634
 Kim, I.S. III-489
 Kim, In-Young I-164
 Kim, Jaegwan IV-152
 Kim, JangSub I-956
 Kim, Jinbae II-977
 Kim, Jin Ho II-1016
 Kim, Jinhwan IV-970
 Kim, Jong G. IV-533
 Kim, Jong Hwa IV-264
 Kim, Juil IV-902
 Kim, Jung-Hyun IV-886
 Kim, Kee-Won III-329, IV-661
 Kim, Ki-Hyung II-1073
 Kim, LaeYoung I-1013
 Kim, Mihui II-1024
 Kim, Mingon IV-152
 Kim, Moonseong I-30, I-38
 Kim, Myungho I-1059
 Kim, Sanghun II-961
 Kim, Seki I-30, I-38
 Kim, Seongbaeg IV-962
 Kim, Soo-Kyun II-279
 Kim, Sung-Ryul IV-289
 Kim, Sun I. I-164
 Kim, Sun-Jeong II-279, II-326
 Kim, Sun Yong II-961
 Kim, Tael I-969
 Kim, Tae-Sun IV-862
 Kim, Tae-Wan IV-280
 Kim, Ung Mo IV-356
 Kim, Won-Sik III-648
 Kirby, R.M. II-147
 Kirley, Michael III-248
 Kisiel-Dorohinicki, Marek III-831,
 III-839, III-908
 Kitowski, Jacek IV-252
 Kleijn, C.R. II-10
 Klein, Dino III-608
 Klie, Hector III-384
 Knight, D. III-473
 Knüpfer, Andreas II-526
 Ko, Young-Bae II-1097
 Kobayashi, Hidetsune I-924
 Kolaczek, Grzegorz III-224
 Kolberg, S. IV-607
 Kong, Xiaohong I-514
 Koo, Jahwan I-948
 Korkhov, Vladimir V. I-530
 Korpeoglu, Ibrahim II-996
 Kosloff, Todd J. II-215
 Köstler, H. II-185
 Kotsokalis, C.A. III-956
 Kotulski, Leszek III-887
 Kou, Gang IV-476, IV-485
 Kowalczyk, W. I-1071
 Kowarz, Andreas IV-541
 Kožaný, Jan III-711
 Kozłak, Jarosław III-703
 Kozłowski, Alex II-215
 Kramer, Robin II-855
 Kranzlmüller, Dieter II-557
 Krawczyk, Henryk I-904
 Krawczyk, M.J. I-665

- Kreaseck, Barbara IV-582
 Krefft, Bogdan I-904
 Kriksciuniene, Dalia IV-316
 Krishnamoorthy, Sriram I-267
 Kroc, Jiří IV-847
 Kryza, Bartosz IV-252
 Krzhizhanovskaya, Valeria V. I-530
 Krznaric, M. III-956
 Kubota, Tetsuyuki II-34
 Küçükosmanoğlu, Alp I-814
 Kujawski, Bernard III-1024
 Kulakowski, K. I-665
 Kulikov, Gennady Yu. I-117, I-781
 Kulikova, Maria V. I-473
 Kulvietienė, Regina II-407
 Kulvietis, Genadijus II-407
 Kurc, Tahsin III-384
 Kurz, Haymo II-86
 Küster, Uwe I-196
 Kwak, Jong Wook IV-979
 Kwoh, Chee Keong II-718
 Kwon, Jeong Ok I-977
 Kwon, Tai-Gil II-1041
 Kwon, Younggoo I-973
- Labatut, Patrick IV-212
 Laclavik, Michal III-980
 Lai, Kin Keung I-790, IV-493
 Lai, Poh-Chin I-884
 Laidlaw, D.H. III-538
 Lambert, T. I-641
 Lambiotte, Renaud III-1114
 Lane, Terran II-895
 Lang, E.W. I-234, I-649
 Langer, Malgorzata I-498
 Larriba-Pey, Josep-L. I-156
 Laskar, Jacques II-446
 Lastovetsky, Alexey III-1008
 Lau, L.C. II-758
 Lavergne, Christian IV-372
 Laws, Joseph IV-870
 Lazarov, Raytcho III-393
 Ledoux, Veerle IV-716
 Lee, Bong-Keun I-908
 Lee, Chilgee II-1089
 Lee, Dongeun IV-922
 Lee, Dong Hoon I-977
 Lee, Doo-Soo I-164
 Lee, Hakjoo II-1065
 Lee, Hanku IV-272, IV-293
- Lee, Hongseok I-960
 Lee, Hyeon-Seok II-1041
 Lee, Hyewon K. I-932
 Lee, Hyongwoo IV-168
 Lee, Hyukjoon IV-930
 Lee, Hyungkeun IV-954
 Lee, Hyung Su IV-910
 Lee, Inbok IV-289
 Lee, J. III-425
 Lee, Jaemyoung I-443, I-761
 Lee, Jae-Woo IV-264
 Lee, Jae Yong II-1033
 Lee, Joo-Haeng II-362
 Lee, Joowan II-977
 Lee, Ju-Hong I-63
 Lee, Junghoon I-985, IV-962, IV-970
 Lee, Junguck II-1065
 Lee, Jysoo I-1059
 Lee, Kang Woong I-108
 Lee, Keon-Myung I-908
 Lee, Kwangyong IV-902
 Lee, Kwan H. II-371
 Lee, Kyu Min III-895, III-899
 Lee, Sang-Ho I-908
 Lee, Sangkeon I-1059, III-972
 Lee, SeoungYoung I-1018
 Lee, Seung-Heon III-200
 Lee, Seung-Jun I-952
 Lee, Seung-Que IV-168
 Lee, Sooyoung II-1065
 Lee, SuKyoung I-1013, II-985
 Lee, Sung-Hee II-1097
 Lee, Sung-Woon I-634
 Lee, Tae-Jin I-989, II-1041
 Lee, Tong-Yee II-295
 Lee, William III-964
 Lee, Woojin IV-902
 Lee, Yeung-Hak IV-862
 Lee, Yuqin IV-805
 Lees, Janet M. I-834
 Lefeuve-Mesgouez, Gaëlle II-50
 Leonard II, J. III-425
 Lepp, Dmitri I-928
 Leshchinskiy, Roman II-920
 Leupi, Célestin II-58
 Levrat, B. I-603
 Lew, A.J. III-456
 Lewis, Andrew I-720
 Li, Bigang III-656, III-687
 Li, Deng III-522

- Li, Dongdong I-435
 Li, Donghai IV-645
 Li, Guoqing I-880, III-17
 Li, Hong IV-918
 Li, Hongxing IV-404
 Li, Jianping IV-501
 Li, Jianyu I-997
 Li, Jianzhong II-662
 Li, Jingtao I-896
 Li, Kuan-Ching I-900
 Li, Liang III-988
 Li, Minglu III-948
 Li, Tao I-250
 Li, Wei III-522
 Li, Wen-hui II-223
 Li, Xiaowei II-1069
 Li, Xiaowen III-9
 Li, Xing IV-176
 Li, Yanhui IV-95
 Li, Yiming IV-599
 Li, Yin I-818
 Li, Ying I-1055
 Li, Yong IV-73, IV-87
 Li, Yuan III-440
 Li, Z. III-554
 Li, Zhenhua IV-404
 Li, Zhong II-358
 Liao, Sheng-hui IV-839
 Liao, Xiaofei IV-380
 Liatsis, P. III-767
 Lie, Knut-Andreas IV-220
 Liljeros, Fredrik III-1048
 Lim, Azman Osman IV-9
 Lin, Chao-Hung II-295
 Lin, Chuang IV-41
 Lin, Po-Feng IV-184
 Lin, Woei IV-49
 Lin, Yongmin III-216
 Lin, Yu IV-128
 Ling, Yun III-184
 Linkens, D.A. I-993
 Lipscomb, William H. IV-533
 Lisik, Zbigniew I-498
 Little, L. II-169
 Liu, Bo III-593
 Liu, Chia-Lung IV-49
 Liu, Dingsheng I-880, III-17
 Liu, Fei I-818, III-1004
 Liu, Feng II-686
 Liu, Jia I-449
 Liu, Jing I-514
 Liu, Kun III-695
 Liu, Ming I-1030
 Liu, Wei II-646
 Liu, Weijiang I-1022, IV-120, IV-144
 Liu, Xiaojian I-769
 Liu, Yang IV-188
 Liu, Zhaodong IV-781
 Liu, Zhiyu IV-404
 Lloret, I. I-316
 Lluch, Javier II-287, II-310
 Lo, Shih-Ching I-1038
 Lodder, Robert J. III-393
 Loh, Woong-Kee III-648
 Loitière, Yannick III-570
 López, Antonio I-555
 López-Ruiz, R. III-353
 Lu, Dongming IV-57
 Lu, Feng I-884
 Lu, Huimei IV-81
 Lu, Jianjiang IV-95
 Lu, Qingda I-267
 Lu, Ssu-Hsuan I-900
 Lu, Yijuan II-686
 Lucas, Philipp IV-200
 Lumbreras, Felipe I-555
 Luo, Fei IV-380
 Luo, Ying I-292, I-876, III-1
 Luque, Emilio I-539
 Lursinsap, Chidchanok II-838
 Lv, Rui I-997
 Lv, Song IV-396
 Ma, Fanyuan I-818, III-1004
 Ma, Guangsheng IV-645
 Ma, Jixin II-775, IV-781
 Ma, Lizhuang II-358
 Ma, Min I-769
 Ma, Xi-min I-826
 Ma, Yan I-880
 Madey, Gregory R. III-417
 Mahfouf, M. I-993
 Mahinthakumar, Kumar III-401
 Maik, Vivek IV-922
 Makowiec, Danuta III-256
 Malaschonok, Gennadi II-486
 Maliekal, J. II-169
 Malinowski, Krzysztof III-791
 Malkowski, Konrad I-242
 Mamat, Ali IV-1071

- Manceny, Matthieu III-1056
 Mandel, Alan K. III-522
 Mandel, Jan III-522
 Măndoiu, I.I. II-758
 Măndoiu, Ion I. II-742
 Manfroi, Fairus I-68
 Manzoni, Sara III-289
 Mao, Zhihong II-358
 Marchese, Fabio M. III-264
 Marchiori, E. I-1071
 Marcjan, Robert III-775
 Margalef, Tomàs I-539
 Marín, Mauricio I-611
 Marroquín-Alonso, Olga II-207
 Martínez, A. IV-685
 Martins, Ana Mafalda II-255
 Martyniak, J. III-956
 Marzo, Jose L. IV-136
 Maskell, Douglas L. I-522
 Masteika, Saulius IV-332
 Mastroianni, Carlo IV-1047
 Mat Deris, M. IV-1071
 Mateja-Losa, Elwira I-24
 Matéo-Vélez, Jean-Charles II-1
 Matossian, Vincent III-384
 Matsukubo, Jun III-1106
 Matsumoto, Noriko IV-436
 Mauch, Sean P. II-122
 Maurizio, Marchese I-547
 Mavridis, Pavlos II-271
 McCalley, James D. III-440
 McCourt, Frederick R.W. II-193
 McCrindle, Rachel I-868
 McGough, A.S. III-956
 McGough, A. Stephen III-964
 McInnes, Lois Curfman I-242
 Meeker, William Q. III-440
 Mei, Jian IV-669
 Meiron, Daniel I. II-122
 Mellema, Angela III-433
 Melnik, Roderick V.N. II-114
 Memon, Ashraf III-920
 Meng, Yu II-223
 Merelli, Emanuela III-1012
 Merkevicius, Egidijus IV-364
 Merkulov, Arkadi I. I-117
 Merschmann, Luiz II-863
 Merzky, Andre III-97
 Mesgouez, Arnaud II-50
 Metaxas, D. III-554
 Michener, William K. III-912
 Michopoulos, John G. II-131, III-456
 Mieke, Philipp III-120
 Mihaylova, Lyudmila III-624
 Miklaszewski, Wiesław III-256
 Milhous, Wilbur K. I-387
 Milledge, Tom II-694, II-702
 Miller, Gavin S.P. IV-228
 Milthorpe, Josh I-218
 Min, Jun-Ki I-364
 Min, Yong I-140, III-1032
 Mingarelli, Angelo B. III-360
 Mishra, Bud II-638, II-654
 Mitrouli, Marilena II-399
 Mix, Hartmut II-526
 Mo, Hongwei I-997
 Mochena, M.D. III-128
 Möller, Kim II-565
 Monfroy, E. I-641
 Monnier, J. II-26
 Moon, Jongbae I-1059
 Moon, Sanghoon I-276, I-284
 Moreira, José E. I-2
 Moreno, A. I-316
 Moreno-Vozmediano, Rafael IV-1031
 Morimoto, Shoichi IV-797
 Morisse, Karsten II-565
 Morley, Chris T. I-834
 Morvan, Michel III-321
 Mou, Tai-yong IV-452
 Mould, David II-318
 Mu, Fei III-687
 Mukherjee, Joy I-46
 Muldoon, C. III-727
 Mun, Sung-Gon I-960
 Munagala, Kamesh III-409
 Muñoz Masqué, J. II-438
 Muñoz, Miguel A. III-1075
 Munoz, Roberto III-1091
 Muntés-Mulero, Victor I-156
 Murugesan, K. I-457
 Murugesu, V. I-457
 Nagar, Atulya I-802
 Nagel, Wolfgang E. II-526
 Najim, M. I-697
 Namachchivaya, N.S. III-448
 Namiki, Takefumi II-34
 Nandigam, Viswanath III-920
 Narasimhan, Giri II-694, II-702, II-807

- Narayanan, Babu I-16
 Nassif, Nabil R. I-148
 Natvig, Jostein R. IV-220
 Naumov, Maxim I-258
 Navarro, Gonzalo I-611
 Navas-Delgado, Ismael III-936
 Nawarecki, Edward III-839
 Nedjalkov, Mihail III-616
 Nenortaitė, Jovita I-1034
 Neves, José III-240
 Ng, Kam-Wing IV-1007
 Nguyen, Ngoc Thanh III-208, III-224
 Nichols, Daniel A. I-387
 Nicole, Denis A. III-928
 Nilsson, Patric II-879
 No, Jaechun IV-1063
 Noël, Alfred G. II-422
 Nooijen, Marcel I-267
 Norris, Boyana I-242
 Nou, Ramon I-84
 Novák, Stanislav I-806
 Nowak, Leszek I-300
 Nutaro, James J. IV-814
 Nygaard, Jens Olav IV-204
- Oberg, Carl III-514
 Oden, J.T. III-530
 O'Grady, M.J. III-727
 O'Hare, G.M.P. III-727
 Oh, Donsung II-977
 Oh, Hyukjun IV-991
 Oh, Jai-Boo IV-661
 Oh, Seungtak II-1089
 Oladunni, Olutayo O. I-188
 Olanda, R. III-13
 Oliveira, Rafael Sachetto I-68, I-76
 Oliveira, S. II-726
 Oliver, Timothy F. I-522
 Olman, Victor II-855
 Olsson, Björn II-879
 Osguthorpe, David I-308
 Ospina, Juan I-920
 Ould-Khaoua, M. I-744
 Overbye, T.J. III-448
 Ozaki, T. I-132
- Pace, Brigida IV-724
 Pachter, R. I-372
 Pai, M.A. III-448
 Paik, Joonki IV-922
- Paik, Juryon IV-356
 Paik, Woojin IV-894
 Pajarola, Renato B. II-371
 Pak, Jinsuk IV-180
 Palekar, M. III-425
 Palkow, Mark II-1050
 Pan, Gang I-435
 Pan, Xuezheng IV-156
 Pan, Yi II-646, II-710
 Pang, Wei II-223
 Papaioannou, Georgios II-271
 Papavassiliou, Dimitrios V. I-188
 Papini, Alessandra IV-677
 Paprzycki, Marcin III-208
 Parashar, Manish III-384
 Parasuk, Vudhichai III-136
 Parasuk, Waraporn III-136
 Park, Chang Won IV-1063
 Park, DongGook III-232
 Park, Geunyoung IV-946
 Park, Gwitae IV-987
 Park, Gyung Leen I-985, II-587, IV-962
 Park, Hong-Shik I-1018
 Park, Hyungjun II-362
 Park, Jaehyung I-969
 Park, Kisoeb I-30, I-38
 Park, Kiyong I-936, II-1081
 Park, Namhun IV-168
 Park, Neungsoo IV-244
 Park, Sangjoon III-1040
 Park, Sang Soon I-944
 Park, SeongHoon I-736
 Park, Sung Soon IV-1063
 Park, Taehyung II-992
 Park, Taesoon III-807
 Park, Tae-Su I-63
 Parker, Steven G. III-393
 Passalis, Georgios II-271
 Pasztor, Egon II-215
 Paszynski, Maciej III-751
 Paternoster, Beatrice IV-700
 Patist, J.P. I-1071
 Paul, Samit I-16
 Paventhan, A. III-928
 Pazo-Robles, M.E. III-337
 Peachey, Tom I-720
 Pecheanu, Emilia II-199
 Pei, Pengjun II-734
 Peláez, Ignacio I-872
 Peng, Bo I-140, III-1032

- Peng, Yanbing I-1022, IV-120
 Peng, Yi IV-476, IV-485
 Pennington, Deana D. III-912
 Pereira, António II-454
 Perelman, Alex II-215
 Pérez, Mariano III-13
 Perumalla, Kalyan II-41
 Peterson, Janet II-177
 Phipps, Eric T. IV-525
 Pickin, Simon IV-765
 Pieczykolan, Jan IV-252
 Pieczynska, Agnieszka III-224, III-891
 Ping, Lingdi IV-156
 Pisa, Ivan T. I-1005
 Pitera, Jed II-846
 Pitman, Michael C. II-846
 Pitzer, Russell M. I-267
 Pivkin, I.V. III-538
 Plastino, Alexandre II-863
 Plaza, Antonio I-888, III-24
 Plaza, Javier III-24
 Pokrywka, Rafał III-855
 Politi, T. IV-708, IV-732
 Pons, Jean-Philippe IV-212
 Popolizio, M. IV-708
 Posse, C. II-871
 Pota, Szabolcs I-830
 Pouchard, Line C. IV-814
 Prank, Rein I-928
 Primavera, Leonardo I-465
 Príncipe, José III-546
 Prăjescu, Claudia II-742
 Prudhomme, S. III-530
 Przekwas, Andrzej IV-822
 Przytycka, Teresa M. II-620
 Pugliese, A. IV-732
 Puglisi, Giovanni II-334
 Puntonet, Carlos G. I-234, I-316, I-356,
 I-649
 Pyle, David Leo I-403
- Qian, Jixin III-593
 Qian, Liang II-904
 Qiao, Daji III-440
 Qin, Guan III-393, III-522
 Qiu, Shibin II-895
 Qu, Youli III-216
 Queiruga Dios, A. II-438
 Quirós, Ricardo II-310
- Rafe, Vahid III-578
 Raghavan, Padma I-242
 Rahmani, Adel Torkaman III-578
 Rajasekaran, Sanguthevar II-822
 Rajasethupathy, K. II-169
 Ramakrishnan, Naren I-46
 Ramalingam, M. III-143
 Ramanujam, J. I-267
 Ramasami, K. III-143
 Ramasami, Ponnadurai III-153
 Ramírez, J. I-234, I-356, I-395, I-649
 Ramos, J.I. II-106
 Ramos, Luis IV-582
 Ramsamy, Priscilla II-595, II-603
 Rangel-Kuoppa, Risto II-318
 Ranjithan, Ranji III-401
 Rasmussen, Craig E. II-945
 Rasúa, Rafael A. Trujillo I-324
 Ravela, Sai III-497
 Ravi, R. II-799
 Raychaudhuri, Dipankar IV-930
 Rayshubskiy, Aleksandr II-846
 Redaelli, Stefano III-289
 Regensburg, Henrik II-1050
 Reis, Artur E. I-842
 Rendell, Alistair P. I-218, II-155
 Reynolds, Paul III-570
 Rhee, Seung Hyong IV-930
 Rhymend Uthariaraj, V. IV-388
 Richard, Adrien II-887
 Richardson, P.D. III-538
 Ridge, Oak II-41
 Ridley, A. III-489
 Riensche, R. II-871
 Rigopoulos, Stelios II-18
 Rizzi, Romeo II-783
 Roberts, Ronald A. III-440
 Roch, Jean-Louis IV-999
 Rodgers, G.J. III-1024
 Rodrigues, Rosália II-454
 Rodríguez, D. IV-789
 Rogier, Francois II-1
 Roh, Yong-Wan IV-886
 Rojek, Gabriel III-823, III-855
 Roman, Eric II-215
 Romero, L.F. II-106
 Romero, Sergio II-912
 Ronald, Nicole III-248
 Ronchieri, E. III-956
 Rong, Chunming I-794, IV-693

- Ros, E. II-518
 Rossello, Damiano IV-324
 Rossman, T. III-473
 Rountev, Atanas I-267
 Rouquier, Jean-Baptiste III-321
 Roux, Olivier II-887
 Roy Mahapatra, Debiprosad II-114
 Rubio, Bartolomé II-912
 Rüde, U. II-185
 Ruiz, R. IV-789
 Russell, A. II-758
 Ryan, J. II-66
 Ryan, Sarah M. III-440
 Ryba, Przemyslaw I-100
 Rylander, M.N. III-530
 Ryu, Jeha II-610
 Ryu, Seungwan II-977, II-1033, IV-168

 Sadayappan, P. I-267
 Safaei, F. I-744
 Sagianos, E. I-798
 Sahingoz, Ozgur Koray III-192, III-903
 Saidane, Mohamed IV-372
 Sakalauskas, Virgilijus IV-316
 Sakurai, Kouichi I-977
 Salinas, Luis I-856
 Saltenis, Vydunas I-704
 Saltz, Joel III-384
 Sameh, Ahmed III-465
 Sánchez, J.R. III-353
 Sanchez, Justin C. III-546
 Sanchez, Maribel II-686
 Sancho Chust, S. II-350
 Sandu, Adrian I-712, III-120, IV-550
 Sanfilippo, A. II-871
 Sankoff, David II-791
 Santana, Miguel IV-999
 Santini, Cindy I-379
 Sappa, Angel I-555
 Sappa, Angel D. I-563
 Sarin, Vivek I-92
 Saubion, F. I-603, I-641
 Sauer, P.W. III-448
 Sautois, B. II-391
 Sbert, Mateu II-263
 Schaefer, Robert III-783, III-799
 Schaubschläger, Christian II-557
 Schloissnig, Siegfried II-502
 Schmidt, Bertil I-522
 Schmidt, Thomas C. II-1050

 Schneider, Adrian II-630
 Schöbel, Rainer IV-340
 Schwan, K. III-425
 Schwartz, Jacob T. II-654
 Schwartz, Russell II-799
 Seber, Dogan I-379
 Sedighian, Saeed III-578
 Segal, Cristina I-172, II-199
 Segura, J.C. I-356, I-395
 Semoushin, Innokenti V. I-473
 Seok, S.C. II-726
 Sequeira, Adélia II-78
 Serrat, Joan I-555
 Seshasayee, B. III-425
 Sethuraman, V. III-143
 Sevinc, Bahadir IV-638
 Sfarti, Dr. Adrian II-215
 Shafi, Aamir II-953
 Shahinpoor, Moshen II-131
 Shakhov, Vladimir V. I-948
 Shang, Wenqian III-216
 Sharifi, Mohsen I-981
 Sharma, Abhishek III-514
 Shi, Hanxiao III-184
 Shi, Yong IV-452, IV-476, IV-485,
 IV-509
 Shi, Zhong-ke IV-878
 Shi, Zhongzhi I-777
 Shiffler, D.A. I-372
 Shigezumi, Takeya II-815
 Shih, Wen-Chung I-810
 Shim, Choon-Bo III-232
 Shin, Chang-Sun III-232
 Shin, Dongchun II-977
 Shin, DongRyeol I-956, III-895, III-899
 Shin, HoJin I-956
 Shin, In-Hye I-985
 Shin, Jeongho IV-922
 Shin, Jitae I-952, IV-25
 Shindin, Sergey K. I-781
 Shirayama, Susumu III-1063
 Shu, Jiwu III-663, III-687, III-695
 Sicilia, M.A. IV-789
 Simha, Rahul III-679
 Simutis, Rimvydas IV-332
 Siwik, Leszek III-831, III-871
 Słota, Damian I-24, I-786
 Ślusarczyk, Grażyna III-883
 Smetek, Marcin II-549
 Smoliy, Elena F. III-879

- Smolka, Maciej III-799
 Śnieżyński, Bartłomiej III-703, III-759
 So, Won-Ho III-232
 Soler, Enrique II-912
 Son, Hyung-Jin III-506
 Son, Jeongho IV-180
 Song, Byunghun IV-910
 Song, Chang-Geun II-326
 Song, Hyoung-Kyu II-969, II-1058
 Song, In-Ho I-164
 Song, JooSeok I-1013, II-1101
 Song, Jungwook I-936, II-1081
 Song, Mingli I-449
 Song, Minseok I-1075
 Song, Sung Keun II-587
 Song, Yong Ho IV-244
 Song, Young Seok III-105, III-113
 Song, Zhanjie I-427, I-822
 Sonmez, A. Coskun III-192, III-903
 Soofi, M.A. III-48
 Sosonkina, Masha I-54
 Sottile, Matthew J. II-945
 Soukiassian, Yeran I-148
 Spezzano, Giandomenico IV-1047
 Spiegel, Michael III-570
 Sreepathi, Sarat III-401
 Sridhar, Srinath II-799
 Srinivasan, Kasthuri I-92
 Srovnal, Vilém III-711
 Stadlthanner, K. I-234
 Stafford, R.J. III-530
 Stagni, Rita IV-831
 Stavrou, Pavlos II-271
 Stauffer, Beth III-514
 Stefanescu, Diana II-199
 Strout, Michelle Mills IV-574, IV-582
 Su, Fanjun IV-156
 Su, Hui-Kai IV-184
 Su, Sen IV-73, IV-87, IV-104, IV-164
 Su, Xianchuang I-140, III-1032
 Sudholt, Wibke III-69
 Suh, Jonghyun II-1065
 Suits, Frank II-846
 Sukhatme, Gaurav III-514
 Sun, Bing II-654
 Sun, Jizhou I-419
 Sun, Jun III-847
 Sunderam, Vaidy I-1
 Sung, Hocheol IV-260
 Susitaival, Riikka IV-420
 Sussman, A. III-448
 Škvor, Jiří I-806
 Šuvakov, Milovan III-1098
 Švec, Martin I-806
 Swiecicki, Mariusz I-300
 Swierszcz, Pawel II-534
 Swope, William II-846
 Sygkouna, Irene I-892
 Sykas, Efstathios I-892
 Szabo, Gabor III-417
 Szczerba, Dominik II-86
 Székely, Gábor II-86
 Szychowiak, Michał I-753
 Tabakow, Iwan III-168
 Tabik, S. II-106
 Tadić, Bosiljka III-1016, III-1024,
 III-1098
 Tahar, M. II-169
 Taherkordi, Amirhosein I-981
 Tai, Phang C. II-710
 Takahashi, Tadashi I-924
 Takaoka, Tadao I-595
 Takeda, Kenji III-928
 Taleghan, Majid Alkaee I-981
 Tan, Feng II-678
 Tan, X.G. IV-822
 Tang, Kai II-342
 Tao, Jie II-502
 Tawfik, H. III-767
 Tawfik, Hissam I-802, III-60
 te Boekhorst, Rene III-367
 Tembe, B.L. III-161
 Theodoropoulos, Georgios III-562
 Theoharis, Theoharis II-271
 Therón, Roberto III-32
 Thivet, Frédéric II-1
 Thomas, Sunil G. III-384
 Thurner, Stefan III-1016, III-1067
 Tian, Jing IV-412, IV-428
 Tian, Qi II-686
 Tibiletti, Luisa IV-324
 Tinnungwattana, Orawan II-838
 Tiyyagura, Sunil R. I-196
 Tong, Ruo-feng IV-839
 Toporkiewicz, W. III-783
 Torosantucci, L. IV-460
 Torres, Jordi I-84
 Tošić, Predrag T. III-272

- Trafalis, Theodore B. I-188, III-506
 Trappe, Wade IV-930
 Trapp, John I-490
 Trebacz, Lechoslaw II-549
 Trelles, Oswaldo III-936
 Triantafyllou, Dimitrios II-399
 Trinh, Thanh Hai IV-1
 Troya, José M. II-912
 Tsai, Ming-Hui II-1008
 Tsechpenakis, G. III-554
 Tseng, Shian-Shyong I-810
 Tsukerman, Igor I-54
 Tu, Shiliang I-1043
 Turek, Wojciech III-775
 Turias, I. I-649
 Tučnák, Petr III-711
 Tuzun, R. II-169
- Uber, Jim III-401
 Uchida, Makoto III-1063
 Ufuktepe, Ünal I-916
 Uhruski, P. III-783
 Ülker, Erkan II-247
 Uribe, Roberto I-611
 Urra, Anna IV-136
- Vaidya, Sheila IV-188
 Vaiksaar, Vahur I-928
 Valaitytė, Akvilina IV-348
 Valakevičius, Eimutis IV-348
 Valencia, David I-888, III-24
 van Bloemen Waanders, Bart III-481
 Van Daele, Marnix IV-716
 Vanden Berghe, Guido IV-716
 Van Hamont, John E. I-387
 Vanneschi, Leonardo III-289
 van Veldhuizen, S. II-10
 Varadarajan, Srinidhi I-46
 Vazirani, V.V. II-758
 Venuvanalingam, P. III-143
 Versteeg, Roelof III-384
 Vialette, Stéphane II-622, II-783
 Vianello, M. IV-685
 Vianna, Gizelle Kupac I-842
 Vidal, Antonio M. I-324, I-348
 Virtamo, Jorma IV-420
 Visscher, Lucas III-97
 Viswanathan, M. III-200
 Vivó, Roberto II-310
 Vodacek, Anthony III-522
- Volkert, Jens II-557
 von Laszewski, Gregor III-401
 Vuik, C. II-10
- Wählich, Matthias II-1050
 Wais, Piotr I-300
 Wajs, Wieslaw I-300
 Walkowiak, Krzysztof I-618, I-626
 Walther, Andrea IV-541
 Wan, Wei I-292, I-876, III-1
 Wan, Zheng IV-156
 Wang, Chaokun II-662
 Wang, Chien-Lung I-1026
 Wang, Chunshan IV-468
 Wang, Di III-695
 Wang, Fang I-1063, III-671, IV-396
 Wang, Feng III-82
 Wang, Guoping I-411
 Wang, Hanpin III-988
 Wang, Hao II-678, IV-428
 Wang, Heng I-411
 Wang, Hongbo IV-128
 Wang, Hsiao-Hsi I-900
 Wang, Jian I-880
 Wang, Jianqin III-1
 Wang, Kuang-Jui I-900
 Wang, Lin I-728, IV-308, IV-517
 Wang, Pei-rong I-826
 Wang, Qijia IV-938
 Wang, Shaowei I-340
 Wang, Shouyang I-790, IV-308, IV-444,
 IV-493, IV-517
 Wang, Shuanghu I-851
 Wang, Xin IV-9
 Wang, Xueping I-896
 Wang, Yan II-223
 Wang, Yang III-663
 Wang, Yueming I-435
 Wang, Yufeng II-686
 Wang, Zhengfang I-292
 Ward, Richard C. IV-814
 Ward, T.J. Christopher II-846
 Wasson, Glenn I-681
 Weber dos Santos, Rodrigo I-68, I-76
 Wedemann, Roseli S. I-842
 Wei, Anne IV-17
 Wei, Guiyi III-184
 Wei, Guozhi IV-17
 Wei, Hu I-864
 Wei, Huang I-790

- Wei, Wei III-948, IV-57
 Wei, Xue III-656
 Weihrauch, Christian III-632, III-640
 Welch, Samuel I-490
 Wen, Wanzhi I-851
 Wheeler, Mary F. III-384
 Whitlock, P.A. III-608
 Wiczorek, Damian IV-1039
 Wierzbowska, Izabela III-719
 Wilhelm, Reinhard IV-200
 Wittevrongel, Sabine IV-65
 Wojcik, Grzegorz M. II-94
 Wojtowicz, Hubert I-300
 Wong, Kim-Sing IV-260
 Wozny, Janusz I-498
 Wu, Chaolin I-292, I-876, III-1
 Wu, Chin-Chi IV-49
 Wu, Haiping III-656
 Wu, Jianping IV-33
 Wu, Kun IV-33
 Wu, Meiping I-689
 Wu, Song IV-1055
 Wu, Yuanxin I-689
 Wu, Zhaohui I-435, I-1055, IV-918,
 IV-938
 Wu, Zhongqiang IV-748
 Wu, Zhauhui IV-468

 Xia, Peng III-671
 Xia, Yu I-8, I-124
 Xia, Yunni III-988
 Xiang, Quanshuang IV-81
 Xiao, Rui I-1030
 Xie, Jiang IV-669
 Xie, Wen IV-444
 Xie, Xia I-1051
 Xiong, Muzhou IV-1055
 Xu, Anbang II-775
 Xu, Baowen IV-95, IV-748
 Xu, Chunxiang III-988
 Xu, Dong II-855
 Xu, Fuyin I-838
 Xu, HaiGuo IV-293
 Xu, Ke IV-17, IV-33
 Xu, Shanying IV-444
 Xu, Weixuan IV-501
 Xu, Wenbo I-514, I-1043, III-847
 Xu, Xian II-670
 Xu, Ying II-855
 Xu, Zhe I-826

 Xue, Wei I-250, III-663, III-695
 Xue, Xiangyang IV-9
 Xue, Yong I-292, I-876, III-1, III-9

 Yaşar, O. II-169
 Yaghmaee, Mohammad Hossien I-588
 Yaikhom, Gagarine II-929
 Yan, Chung-Ren II-295
 Yang, Chao-Tung I-810
 Yang, Fangchun IV-73, IV-87, IV-164
 Yang, Geng I-794, IV-693
 Yang, Jun III-409
 Yang, Kun II-662
 Yang, Mao IV-412, IV-428
 Yang, Mijeong I-969
 Yang, Shouyuan I-427
 Yang, Shuzhong I-997
 Yang, Wang I-1022
 Yang, Xin-She I-834
 Yang, Xuejun II-904
 Yang, Yingchun I-435
 Yang, Young-Kyu III-200
 Yang, Zhanxin I-997
 Yanami, Hitoshi II-462
 Yangchun, Liang I-547
 Yantır, Ahmet I-916
 Yao, Jialiang III-60
 Yao, Yonglei IV-164
 Yazici, Ali II-375
 Ye, Chaoqun I-769
 Yegül, Umüt I-814
 Yélamos, P. I-356
 Yeo, So-Young II-969
 Yeon, Eunja IV-894
 Yi, Huizhan II-904
 Yi, Sangho IV-946
 Yildirim, Tulay IV-615
 Yim, Soon-Bin II-1041
 Yoo, Chae-Woo I-965, II-510
 Yoo, Chuck IV-160, IV-172
 Yoo, Kee-Young II-1000, III-329,
 IV-661
 Yoon, Eun-Jun II-1000
 Yoon, Jungwon II-610
 Yoon, Seokho II-961
 Yoon, Tae-Sun II-830
 Yoon, Won-Sik II-1073
 Yoshida, Norihiko IV-436
 You, L.H. II-231
 You, Mingyu I-449

- You, Young-Hwan II-969, II-1058
 Youn, Choonhan I-379
 Youn, Hee Yong II-587, IV-1, IV-910
 Youn, Jaehwan III-807
 Yu, Chiu-Man IV-1007
 Yu, Dongjin I-1055
 Yu, Hongliang III-656
 Yu, Jun III-184
 Yu, Lean I-790, IV-308, IV-444,
 IV-493, IV-517
 Yu, Mei II-367
 Yu, Shao-Ming I-1038
 Yu, Shaokai II-1073
 Yu, Zhiping I-250
 Yuan, Ding I-179
 Yuan, Ruifeng IV-404
 Yue, Wuyi IV-509
 Yun, Hyunho IV-152

 Żabińska, Małgorzata III-735
 Zechman, Emily III-401
 Zelikovsky, Alex II-750
 Zelikovsky, Alexander II-767
 Zeng, Lingfang I-1063, III-671, IV-396
 Zeng, Yurong I-728
 Zenger, Ch. I-673
 Zhang, Aidong II-670, II-734
 Zhang, Bin III-514
 Zhang, Chengwen IV-104
 Zhang, Shunda IV-396
 Zhang, Gendu I-896
 Zhang, Guang-Yan III-663
 Zhang, Guiling I-419
 Zhang, Hao IV-380
 Zhang, Jian J. II-231
 Zhang, Jianting III-912
 Zhang, Jinlong I-728
 Zhang, Liguó II-478
 Zhang, Qianli IV-176
 Zhang, Qin I-1051, IV-380
 Zhang, Shensheng III-948
 Zhang, Shijia I-411
 Zhang, Shunda IV-396
 Zhang, Wanjun III-17
 Zhang, Weide I-681

 Zhang, Wenju I-818, III-1004
 Zhang, Wenyi III-17
 Zhang, Wu I-1047, IV-669
 Zhang, Y. III-530
 Zhang, Yanqing II-678
 Zhang, Yingzhou IV-748
 Zhang, Yiyiing II-830
 Zhao, Bo I-657
 Zhao, Chen I-204
 Zhao, Guiping I-851
 Zhao, Guoxing II-775, IV-781
 Zhao, Jun III-593
 Zhao, Liqiang IV-9
 Zhao, Mingxi II-358
 Zhao, Peng III-1008
 Zhao, Wenyun IV-805
 Zheng, Bo IV-41
 Zheng, Chunfang II-791
 Zheng, Gaolin II-694, II-702
 Zheng, Lei I-292, I-876, III-1
 Zheng, Nenggan IV-938
 Zheng, Weimin III-656, III-687
 Zhestkov, Yuri II-846
 Zhong, Shaobo I-292, I-876, III-1, III-9
 Zhong, Wei II-710
 Zhou, Deyu II-718
 Zhou, Mingzhong IV-112
 Zhou, Ruhong II-846
 Zhou, Xingwei I-427, I-822
 Zhou, Yanmiao IV-938
 Zhou, Yi I-1055
 Zhou, Yu II-367
 Zhou, Zhongmei IV-468
 Zhou, Zong-fang IV-452
 Zhu, Haibin III-216
 Zhu, Qiuping I-340
 Zhu, Wei-yong I-826
 Zhuang, Yueting I-204
 Zieliński, Krzysztof III-940, IV-1023,
 IV-1039
 Zory, J. IV-653
 Zou, Deqing IV-1055
 Zuo, Xin III-1008
 Zuzarte, Calisto I-156