

James O'Shea Ngoc Thanh Nguyen
Keeley Crockett Robert J. Howlett
Lakhmi C. Jain (Eds.)

LNAI 6682

Agent and Multi-Agent Systems: Technologies and Applications

5th KES International Conference, KES-AMSTA 2011
Manchester, UK, June/July 2011
Proceedings

Lecture Notes in Artificial Intelligence

6682

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

James O'Shea Ngoc Thanh Nguyen
Keeley Crockett Robert J. Howlett
Lakhmi C. Jain (Eds.)

Agent and Multi-Agent Systems: Technologies and Applications

5th KES International Conference, KES-AMSTA 2011
Manchester, UK, June 29 – July 1, 2011
Proceedings

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

James O'Shea
Keeley Crockett
Manchester Metropolitan University
Manchester, M1 5GD, UK
E-mail: {j.d.oshea,k.crockett}@mmu.ac.uk

Ngoc Thanh Nguyen
Wroclaw University of Technology
50-370 Wroclaw, Poland
E-mail: ngoc-thanh.nguyen@pwr.wroc.pl

Robert J. Howlett
Bournemouth University
Poole BH12 5BB, UK
E-mail: rjhowlett@kesinternational.org

Lakhmi C. Jain
University of South Australia
Mawson Lakes, SA, 5095, Australia
E-mail: lakhmi.jain@unisa.edu.au

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21999-3 e-ISBN 978-3-642-22000-5
DOI 10.1007/978-3-642-22000-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011929847

CR Subject Classification (1998): I.2, H.2.8, H.3-5, J.4, C.2, D.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

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.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the proceedings of the KES Conference on Agent and Multi-Agent Systems – Technologies and Applications (KES-AMSTA 2011) hosted by Manchester Metropolitan University in Manchester between June 29 and July 1, 2011. The conference was organized by Manchester Metropolitan University, KES International and its focus group on agent and multi-agent systems. The KES-AMSTA conference is a subseries of the KES conference series.

Following the successes of previous KES Symposia on Agent and Multi-Agent Systems – Technologies and Applications, held in Wroclaw, Poland (KES-AMSTA 2007), Incheon, Korea (KES-AMSTA 2008), Uppsala, Sweden (KES-AMSTA 2009) and Gdynia, Poland (KES-AMSTA 2010), it was decided to run KES-AMSTA 2011 as a full conference. KES-AMSTA 2011 featured the usual keynote talks, oral presentations and invited sessions closely aligned to the established themes of the conference.

The aim of the conference was to provide an internationally respected forum for scientific research in the technologies and applications of agent and multi-agent systems. This field is concerned with the development and evaluation of sophisticated, AI-based problem-solving and control architectures for both single-agent and multi-agent systems. Current topics of research in the field include (amongst others) agent-oriented software engineering, beliefs, desires and intentions, agent co-operation, co-ordination, negotiation, organization and communication, distributed problem solving, specification of agent communication languages, formalization of ontologies and conversational agents.

The conference attracted a substantial number of researchers and practitioners from all over the world who submitted their papers for eight main tracks covering the methodology and applications of agent and multi-agent systems and five special sessions on specific topics within the field.

Submissions came from 20 countries. Each paper was peer reviewed by at least two members of the International Programme Committee and International Reviewer Board.

The Programme Committee defined the following main tracks: Conversational Agents, Dialogue Systems and Text Processing, Agents and Online Social Networks, Modelling, Planning and Prediction, Robotics and Manufacturing, Agent Optimization, Negotiation and Security, Multi-Agent Systems and Mining and Profiling.

In addition to the main tracks of the conference there were the following five special sessions: Agent-Based Optimization (ABO2011), Computer-Support Social Intelligence for Human Interaction (IS01), Digital Economy (DE 2011) and Intelligent Workflow, Cloud Computing and Systems (IS03) as well as a Doctoral Track. This year's conference also saw the introduction of the New

Directions special session which provides a forum for work that is still at an early stage in development or broadens the theme of the conference.

Accepted and presented papers highlight new trends and challenges in agent and multi-agent research. We hope these results will be of value to the research community working in the fields of artificial intelligence, collective computational intelligence, robotics, dialogue systems and, in particular, agent and multi-agent systems, technologies and applications.

We would like to express our sincere thanks to the Honorary Chair, John Brooks, Vice-Chancellor of Manchester Metropolitan University, UK, for his support, and Gerry Kelleher, Deputy Vice-Chancellor (Strategic Planning) of Manchester Metropolitan University for giving the welcome address.

The Chairs' special thanks go to the local Organizing Committee, chaired by James O'Shea, who did very solid and excellent work. Thanks are due to the Programme Co-chairs, all Programme and Reviewer Committee members and all the additional reviewers for their valuable efforts in the review process, which helped us to guarantee the highest quality of selected papers for the conference.

We would like to thank our main sponsor, Manchester Metropolitan University. Our special thanks also go to Springer for publishing the proceedings.

We would also like to express our thanks to the keynote speakers, Michael Wooldridge and Piotr Jędrzejowicz for their interesting and informative talks of a world-class standard. We cordially thank all of the authors for their valuable contributions and all of the other participants in this conference. The conference would not be possible without their support.

June 2011

James O'Shea
Keeley Crockett
Ngoc Thanh Nguyen
Robert J. Howlett
Lakhmi C. Jain

KES-AMSTA 2011 Conference Organization

Honorary Chair

John Brooks

Vice-Chancellor of Manchester Metropolitan
University, UK

General Chairs

James D. O'Shea

Ngoc Thanh Nguyen

Lakhmi C. Jain

Manchester Metropolitan University, UK

Wroclaw University of Technology, Poland

University of South Australia, Australia

Executive Chair

Robert J. Howlett

KES International and the University of
Bournemouth, UK

Programme Co-chairs

Zuhair Bandar

Keeley Crockett

Dariusz Król

Manchester Metropolitan University, UK

Manchester Metropolitan University, UK

Wroclaw University of Technology, Poland

Special Sessions Chairs

1. Agent-Based Optimization (ABO 2011)

Piotr Jędrzejowicz, Gdynia Maritime University, Poland

2. Computer-Supported Social Intelligence for Human Interaction (IS01)

Toyohide Watanabe, Nagoya University, Japan

3. Intelligent Workflow, Cloud Computing and Systems (IS03)

Yun-Heh (Jessica) Chen-Burger, University of Edinburgh, UK

4. Digital Economy (DE 2011)

Arkadiusz Kawa, Poznań University of Economics, Poland

5. Doctoral Track

Ireneusz Czarnowski, Gdynia Maritime University, Poland

Publicity Co-chairs

D. McLean
K. Crockett

Manchester Metropolitan University, UK
Manchester Metropolitan University, UK

Liaison Chair

T. Watanabe

Nagoya University, Japan

Keynote Speakers

Michael Wooldridge
University of Liverpool, UK
Computational Aspects of Cooperative Game Theory

Piotr Jędrzejowicz
Gdynia Maritime University
Machine Learning and Agents

International Programme Committee

Asakura, Koichi
Azzouzi, Messaouda
Bandar, Zuhair
Barbucha, Dariusz
Basilico, Nicola
Bielikova, Maria

Brzykcy, Grażyna
Čapkovič, František

Cetnarowicz, Krzysztof

Chen-Burger, Yun-Heh
Crispin, Alan
Cristani, Matteo
Crockett, Keeley
Czarnowski, Ireneusz
Davidsson, Paul
Dunin-Kępicz, Barbara
Fernandez-Caballero, Antonio

Gatti, Nicola

Daido University, Japan
Ziane Achour University of Djelfa, Algeria
Manchester Metropolitan University, UK
Gdynia Maritime University, Poland
Politecnico di Milano, Italy
Slovak University of Technology in
Bratislava, Slovakia
Poznań University of Technology, Poland
Institute of Informatics, Slovak Academy of
Sciences, Slovakia
AGH University of Science and Technology,
Poland
The University of Edinburgh, UK
Manchester Metropolitan University, UK
Università di Verona, Italy
Manchester Metropolitan University, UK
Gdynia Maritime University, Poland
Blekinge Institute of Technology, Sweden
University of Warsaw, Poland
University of Castilla-La Mancha, Albacete,
Spain
Politecnico di Milano, Italy

Georgieva, Lila	Heriot-Watt University, Edinburgh, UK
Golińska, Paulina	Poznań University of Technology, Poland
Hakansson, Anne	KTH Royal Institute of Technology, Sweden
Hammoudeh, Mohammed	Manchester Metropolitan University, UK
Hanachi, Chihab	IRIT Laboratory, University of Toulouse, France
Hartung, Ronald L.	Franklin University, USA
Jatowt, Adam	Kyoto University, Japan
Jędrzejowicz, Joanna	University of Gdansk, Poland
Jędrzejowicz, Piotr	Gdynia Maritime University, Poland
Jezic, Gordan	University of Zagreb, Croatia
Jonquet, Clement	University of Montpellier, France
Józefowska, Joanna	Poznań University of Technology, Poland
Jung, Jason J.	Yeungnam University, Korea
Juszczyszyn, Krzysztof	Wroclaw University of Technology, Poland
Katarzyniak, Radosław Piotr	Wroclaw University of Technology, Poland
Kawa, Arkadiusz	Poznań University of Economics, Poland
Kazienko, Przemysław	Wroclaw University of Technology, Poland
Klein, Michel	VU University Amsterdam, The Netherlands
Król , Dariusz	Wroclaw University of Technology, Poland
Kudenko, Daniel	University of York Heslington, UK
Kušek, Mario	University of Zagreb, Croatia
Kuwabara, Kazuhiro	Ritsumeikan University, Japan
Lee, Jaeho	The University of Seoul, Korea
Lin, Fang-Pang	National Center for High-Performance Computing, Taiwan
Lin, Fuhua	Athabasca University, Canada
McLean, David	Manchester Metropolitan University, UK
Moldt, Daniel	University of Hamburg, Germany
Mukai, Naoto	Tokyo University of Science, Japan
Muyeba, Maybin	Manchester Metropolitan University, UK
Nadarajan, Gaya	University of Edinburgh, UK
Nawarecki, Edward	AGH University of Science and Technology, Poland
O'Shea, James	Manchester Metropolitan University, UK
Prasad, Bhanu	Florida A&M University, USA
Raś, Zbigniew	University of North Carolina-Charlotte, USA
Ratajczak-Ropel, Ewa	Gdynia Maritime University, Poland
Reghunadhan, Rajesh	Network Systems and Technologies Ltd., India
Ricci, Alessandro	University of Bologna, Italy
Savarimuthu, Bastin Tony Roy	University of Otago, Dunedin, New Zealand
Seto, Mae L.	Dalhousie University, Canada
Sharpanskykh, Alexei	Vrije Universiteit Amsterdam, The Netherlands

Skakovski, Aleksander
Sobecki, Janusz
Tadeusiewicz, Ryszard

Gdynia Maritime University, Poland
Wroclaw University of Technology, Poland
AGH University of Science and Technology,
Poland

Takama, Yasufumi
Wieczerzycki, Waldemar
Thomas, Wojciech
Ushiana, Taketoshi
Watanabe, Toyohide
Yeh, Ching-Long
Zeng, Daniel

Tokyo Metropolitan University, Japan
Poznań University of Economics
Wroclaw University of Technology, Poland
Kyushu University, Japan
Nagoya University, Japan
Tatung University, Taiwan
Chinese Academy of Sciences and University
of Arizona, USA

Zhang, Wen-Ran
Zieliński, Krzysztof

Georgia Southern University, USA
AGH University of Science and Technology,
Poland

KES-AMSTA Symposium Series and Focus Group on Agent and Multi-Agent System Chair

Ngoc Thanh Nguyen

Wrocław University of Technology, Poland

KES International Conference Series Chairs

Robert J. Howlett
Lakhmi C. Jain

Bournemouth University, UK
University of South Australia, Australia

Table of Contents

Keynote Speeches

Computational Aspects of Cooperative Game Theory (Abstract)	1
<i>Michael Wooldridge</i>	

Machine Learning and Agents	2
<i>Piotr Jędrzejowicz</i>	

Conversational Agents, Dialogue Systems and Text Processing (Gen01)

Goal Orientated Conversational Agents: Applications to Benefit Society	16
<i>Keeley Crockett, James O'Shea, and Zuhair Bandar</i>	

Using a Slim Function Word Classifier to Recognise Instruction Dialogue Acts	26
<i>James O'Shea, Zuhair Bandar, and Keeley Crockett</i>	

An Application of Pattern Matching Stemmer in Arabic Dialogue System	35
<i>Mohammad Hijjawi, Zuhair Bandar, Keeley Crockett, and David Mclean</i>	

Information Extraction for SQL Query Generation in the Conversation-Based Interfaces to Relational Databases (C-BIRD)	44
<i>Majdi Owda, Zuhair Bandar, and Keeley Crockett</i>	

Agents and Online Social Networks (Gen02)

Multi Agent System for Historical Information Retrieval from Online Social Networks	54
<i>Ruqayya Abdulrahman, Daniel Neagu, and D.R.W. Holton</i>	

Social-Network Influence on Telecommunication Customer Attrition	64
<i>Piotr Wojewnik, Bogumil Kaminski, Mateusz Zawisza, and Marek Antosiewicz</i>	

An Agent-Based Platform for Ad-Hoc Social Networking	74
<i>Vedran Podobnik and Ignac Lovrek</i>	

Formal Specification of Multi Agent System for Historical Information Retrieval from Online Social Networks 84
Ruqayya Abdulrahman, D.R.W. Holton, Daniel Neagu, and Mick Ridley

Modelling, Planning and Prediction (Gen03)

A Multi-agent System for the Modelling of the HIV Infection 94
Toufik Laroum and Bornia Tighiouart

A Multi-agents System for Analysis and Diagnosis of SMEs 103
Cecilia Zanni-Merk, Santiago Almiron, and Dominique Renaud

Model of Petri Net for Representing an Agent Plan in Dynamic Environment 113
Said Brahimi, Ramdane Maamri, and Zaidi Sahnoun

Towards a Predictive Fault Tolerance Approach in Multi-Agent Systems 123
Samira Telghamti and Ramdane Maamri

Modelling Three-Phase Power Network with a Multi-Agent System 130
Miroslav Prýmek, Aleš Horák, and Tadeusz Sikora

A Method of User Modeling and Relevance Simulation in Document Retrieval Systems 138
Bernadetta Mianowska and Ngoc Thanh Nguyen

A Multi-agent Organizational Model for Grid Scheduling 148
Inès Thabet, Issam Bouslimi, Chihab Hanachi, and Khaled Ghédira

Extending the JADE Agent Behaviour Model with *JBehaviourTrees* Framework 159
Iva Bojic, Tomislav Lipic, Mario Kusek, and Gordan Jezic

Scenario Description Language in Multi-agent Simulation System 169
Armin Stranjak, Igor Čavrak, and Mario Žagar

Analyzing the Validity of Passive Investment Strategies Employing Fundamental Indices through Agent-Based Simulation 180
Hiroshi Takahashi, Satoru Takahashi, and Takao Terano

Robotics and Manufacturing (Gen04)

Dynamic World Model with the Lazy Potential Function 190
Konrad Kulakowski and Tomasz Stepień

VigilAgent for the Development of Agent-Based Multi-robot Surveillance Systems	200
<i>José M. Gascueña, Elena Navarro, and Antonio Fernández-Caballero</i>	
Searching Targets Using Mobile Agents in a Large Scale Multi-robot Environment	211
<i>Tsukasa Abe, Munehiro Takimoto, and Yasushi Kambayashi</i>	
Cooperation of Hybrid Agents in Models of Manufacturing Systems	221
<i>František Čapkovič</i>	
Multi-Agent Oriented Integration in Distributed Control System	231
<i>Dariusz Choinski and Michał Senik</i>	
pPDC Blackboard Broadcasting in Agent-Based Distributed Process Control	241
<i>Grzegorz Polaków and Mieczysław Metzger</i>	
Agent Optimisation (Gen05)	
Sequence Automata for Researching Consensus Levels	251
<i>Henryk Piech</i>	
Agent-Based Simulation for Equilibrium Selection and Coordination Failure in Minimum Strategy Coordination Games	261
<i>Ichiro Nishizaki, Tomohiro Hayashida, and Noriyuki Hara</i>	
A Genetic Algorithm for Community Formation Based on Collective Intelligence Capacity	271
<i>Emil Scarlat and Iulia Maries</i>	
Negotiation and Security (Gen06)	
Argumentative Agents Negotiating on Potential Attacks	280
<i>Guido Boella, Dov M. Gabbay, Alan Perotti, Leendert van der Torre, and Serena Villata</i>	
Environment-Driven Threats Elicitation for Web Applications	291
<i>Hui Guan, Weiru Chen, Lin Liu, and Hongji Yang</i>	
Towards Ontology of Fraudulent Disbursement	301
<i>Czesław Jędrzejek, Jolanta Cybulka, and Jarosław Bgk</i>	
A New Mechanism for Negotiations in Multi-Agent Systems Based on ARTMAP Artificial Neural Network	311
<i>R. Beheshti and N. Mozayani</i>	

Multi-Agent Systems (Gen07)

A Multi-Agent Architectural Pattern for Wrapping Off-the-Shelf Components	321
<i>Sodany Kiv, Yves Wautelet, and Manuel Kolp</i>	
A Multi-Agent System for Game Trading on the B2B Electronic Market	332
<i>Ana Petric and Gordan Jezic</i>	
Experimental Comparison of Resampling Methods in a Multi-Agent System to Assist with Property Valuation	342
<i>Tadeusz Lasota, Tomasz Luczak, and Bogdan Trawiński</i>	
Voting in Multi-Agent System for Improvement of Partial Observations	353
<i>Dariusz Choinski, Mieczyslaw Metzger, and Witold Nocon</i>	
Active Sets and Unification	363
<i>Germano Resconi and Chris Hinde</i>	
Evaluating Interaction of MAS Providing Context-Aware Services	373
<i>Nayat Sanchez-Pi, David Griol, Javier Carbo, and Jose M. Molina</i>	
Collaborative Workflow Management in Holonic Multi-Agent Systems	383
<i>Fu-Shiung Hsieh</i>	
Agent-Oriented Requirements Analysis from Scenarios	394
<i>Lin Liu, Zhi Jin, Ruqian Lu, and Hongji Yang</i>	

Mining and Profiling (Gen08)

Application of Multi-agents in TV Commercial Recognition System	406
<i>Pawel Biernacki</i>	
A Hybrid Interestingness Heuristic Approach for Attribute-Oriented Mining	414
<i>Maybin Muyebe, Keeley Crockett, and John Keane</i>	

Agent-Based Optimization (ABO 2011)

The Network Balance Realized by Routing Organization System	425
<i>Henryk Piech and Adam Czubak</i>	
A New Cluster-Based Instance Selection Algorithm	436
<i>Ireneusz Czarnowski and Piotr Jędrzejowicz</i>	

An Agent-Based Approach to the Dynamic Price Problem	446
<i>Yuri Chizhov</i>	
Parallel Cooperating A-Teams Solving Instances of the Euclidean Planar Travelling Salesman Problem	456
<i>Piotr Jędrzejowicz and Izabela Wierzbowska</i>	
A-Team for Solving MRCPSP/max Problem	466
<i>Piotr Jędrzejowicz and Ewa Ratajczak-Ropel</i>	
An Agent-Based Guided Local Search for the Capacited Vehicle Routing Problem	476
<i>Dariusz Barbucha</i>	
An Agent-Based Framework for Solving an Equity Location Problem . . .	486
<i>Maria Barbati, Giuseppe Bruno, and Andrea Genovese</i>	
Rotation Forest with GEP-Induced Expression Trees	495
<i>Joanna Jędrzejowicz and Piotr Jędrzejowicz</i>	
Doctoral Track	
Combined Course Programmes Generation in Multi-agent e-Learning System Using Policy-Based HTN Planning	504
<i>Pavel Nikolaev and Aladdin Ayesh</i>	
SimZombie: A Case-Study in Agent-Based Simulation Construction . . .	514
<i>Matthew Crossley and Martyn Amos</i>	
Towards Agent-Based Crowd Simulation in Airports Using Games Technology	524
<i>Olivier Szymanczyk, Patrick Dickinson, and Tom Duckett</i>	
Using Properties to Compare Both Words and Clauses	534
<i>David M. Pearce, Zuhair Bandar, and David McLean</i>	
Semantic Similarity Measures for the Development of Thai Dialog System	544
<i>Khukrit Osathanunkul, James O'Shea, Zuhair Bandar, and Keeley Crockett</i>	
Graph Coloring with a Distributed Hybrid Quantum Annealing Algorithm	553
<i>Olawale Titiloye and Alan Crispin</i>	
Oscar: An Intelligent Adaptive Conversational Agent Tutoring System	563
<i>Annabel Latham, Keeley Crockett, David McLean, and Bruce Edmonds</i>	

Computer-Support Social Intelligence for Human Interaction (IS01 2011)

An Information Recommendation Agent on Microblogging Service 573
Taketoshi Ushiana and Tomoya Eguchi

Movement Algorithms for Refugee Agents for Virtual Disaster Simulation Systems 583
Koichi Asakura, Hiroki Aoyama, and Toyohide Watanabe

Vehicle Routing Based on Traffic Cost at Intersection 592
Toyohide Watanabe and Chang Lijiao

Digital Economy (DE 2011)

Virtual Organization Networking Strategies – Simulations Experiments 602
Konrad Fuks and Arkadiusz Kawa

Multistage Covariance Approach to Measure the Randomness in Financial Time Series Analysis 610
Ryszard Szupiluk, Piotr Wojewnik, and Tomasz Zabkowski

Multi-agent Coordination Mechanism of Virtual Supply Chain 620
Paulina Golińska and Marcin Hajdul

Competence-Based Architecture for Knowledge Logistics in Project-Oriented Organization 630
Przemysław Różewski and Bartłomiej Małachowski

The Effects of Knowledge Management System Quality and Self-efficacy on Intention to Knowledge Sharing: The Mediate Role of Attitude 640
Shiuann-Shuoh Chen, Yu-Wei Chuang, and Pei-Yi Chen

Social Network Analysis as a Tool for Improving Enterprise Architecture 651
Przemysław Kazienko, Radosław Michalski, and Sebastian Palus

Intelligent Workflow, Cloud Computing and Systems (IS03)

Dynamic Workflow Adaptation over Adaptive Infrastructures 661
Rafael Tolosana-Calasanç, José A. Bañares, and Omer F. Rana

ebXML Business-to-Business Integration Based on Semantic Cloud Computing Architecture 671
Ching-Long Yeh

SWAV: Semantics-Based Workflows for Automatic Video Analysis	681
<i>Gayathri Nadarajan, Yun-Heh Chen-Burger, and Robert B. Fisher</i>	
A Meta-Model of Business Interaction for Assisting Intelligent Workflow Systems	692
<i>Areti Manataki and Yun-Heh Chen-Burger</i>	
Author Index	703

Computational Aspects of Cooperative Game Theory

Michael Wooldridge

Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK
mjw@liv.ac.uk

Abstract. The theory of cooperative games provides a rich mathematical framework with which to understand the interactions between self-interested agents in settings where they can benefit from cooperation, and where binding agreements between agents can be made. Our aim in this talk is to describe the issues that arise when we consider cooperative game theory through a computational lens. We begin by introducing basic concepts from cooperative game theory, and in particular the key solution concepts: the core and the Shapley value. We then introduce the key issues that arise if one is to consider the cooperative games in a computational setting: in particular, the issue of representing games, and the computational complexity of cooperative solution concepts.

Machine Learning and Agents

Piotr Jędrzejowicz

Chair of Information Systems, Gdynia Maritime University,
Morska 83, 81-225 Gdynia, Poland
pj@am.gdynia.pl

Abstract. The paper reviews current research results integrating machine learning and agent technologies. Although complementary solutions from both fields are discussed the focus is on using agent technology in the field of machine learning with a particular interest on applying agent-based solutions to supervised learning. The paper contains a short review of applications, in which machine learning methods have been used to support agent learning capabilities. This is followed by a corresponding review of machine learning methods and tools in which agent technology plays an important role. Final part gives a more detailed description of some example machine learning models and solutions where the asynchronous team of agents paradigm has been implemented to support the machine learning methods and which have been developed by the author and his research group.

1 Introduction

Contemporary definition sees machine learning as a discipline that is concerned with the design and development of algorithms that allow computers to learn behaviors based on empirical data. Data can be seen as examples that illustrate relations between observed objects. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data. Parallel to recent developments in the field of machine learning, mainly as a result of convergence of many technologies within computer science such as object-oriented programming, distributed computing and artificial life, the agent technology has emerged. An agent is understood here as any piece of software that is designed to use intelligence to automatically carry out an assigned task, mainly retrieving and delivering information. Tweedale and co-authors [62] outline an abridged history of agents as a guide for the reader to understand the trends and directions of future agent design. This description includes how agent technologies have developed using increasingly sophisticated techniques. It also indicates the transition of formal programming languages into object-oriented programming and how this transition facilitated a corresponding shift from scripted agents (bots) to agent-oriented designs which is best exemplified by multiple agent systems (MAS). A MAS tries to solve complex problems with entities called agents, using their collaborative and autonomous properties [38]. Basic MAS properties are listed in [34].

During the last decade developments in the fields of machine learning and agent technologies have, in some respect, become complementary and researchers from both fields have seen ample opportunities to profit from solutions proposed by each other. Several agent-based frameworks that utilize machine learning for intelligent decision support have been recently reported. Learning is increasingly being seen as a key ability of agents, and research into learning agent technology, such as reinforcement learning and supervised or unsupervised learning has produced many valuable applications. In this paper the focus is however on using agent technology in the field of machine learning with a particular interest on applying agent-based solutions to supervised learning. Supervised learning is the machine learning task of inducing a function from training data which is a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal or class label). A supervised learning algorithm analyzes the training data and produces an induced function, which is called a classifier if the output is discrete, or a regression function if the output is continuous. The inferred function should predict the correct output value for any valid input object. This requires the learning algorithm to generalize from the training data to unseen situations.

There are several ways the machine learning algorithm can profit from applying agent technology. Among them the following will be addressed in this paper:

- There are numerous machine learning techniques where parallelization can speed-up or even enable learning. Using a set of agents may, in such circumstances, increase efficiency of learning.
- Several machine learning techniques directly rely on the collective computational intelligence paradigm, where a synergetic effect is expected from combining efforts of various program agents.
- There is a class of machine learning problems known as the distributed machine learning. In the distributed learning a set of agents working in the distributed sites can be used to produce some local level solutions independently and in parallel. Later on local level solutions are combined into a global solution.

The paper is organized as follows. Section 2 contains a short review of applications, in which machine learning methods have been used to support agent learning capabilities. Section 3 offers a corresponding review of machine learning methods and tools in which agent technology plays an important role. Section 4 gives more detailed description of some example machine learning models and solutions where the agent paradigm has been implemented and which have been developed by the author and his research group. Finally, conclusions contain suggestions on future research and possible deeper integration of machine learning and agent technology.

2 Learning Agents

Probably the most often used approach to provide agents with learning capabilities is the reinforcement learning. An excellent survey of multiagent reinforcement learning can be found in [10]. As it was pointed out by Sutton and Barto [57] reinforcement learning is learning what to do - how to map situations to actions so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward. In describing properties of the reinforcement learning the authors directly refer to the notion of agent. In their view a learning agent must be able to sense the state of the environment and must be able to take actions that affect the state. The agent also must have goal or goals relating to the state of the environment [57].

Theoretical developments in the field of learning agents focus mostly on methodologies and requirements for constructing multiagent systems with learning capabilities. Connection of the theory of automata with the multiagent reinforcement learning is explored in [45]. Shoham et.al. [53] claim that the area of learning in multi-agent systems is today one of the most fertile grounds for interaction between game theory and artificial intelligence. In [41] challenges motivated by engineering applications and the potential appeal of multi-agent learning to meet these challenges are discussed.

Symeonidis et.al. [58] present an approach that takes the relevant limitations and considerations into account and provides a gateway on the way data mining techniques can be employed in order to augment agent intelligence. This work demonstrates how the extracted knowledge can be used for the formulation initially, and the improvement, in the long run, of agent reasoning. Preux et.al. [47] present MAABAC, a generic model for building adaptive agents: they learn new behaviors by interacting with their environment. Agents adapt their behavior by way of reinforcement learning, namely temporal difference methods. The paper [52] presents a systematic approach to introduce machine learning in the design and implementation phases of a software agent. It also presents an incremental implementation process for building asynchronous and distributed agents, which supports the combination of machine learning strategies. Rosaci in [51] proposes a complete MAS architecture, called connectionist learning and inter-ontology similarities (CILIOS), for supporting agent mutual monitoring.

In [42] the concepts of stigmergy and entropy are imported into learning automata based multi-agent systems with the purpose of providing a simple framework for interaction and coordination in multi-agent systems and speeding up the learning process. Another extension was proposed in [8]. The authors suggest a merging, and hence an extension, of two recent learning methods, utility-based learning and strategic or adversarial learning. Utility-based learning brings to the forefront the learner's utility function during induction. Strategic learning anticipates strategic activity in the induction process when the instances are intelligent agents such as in classification problems involving people or organizations. The resulting merged model is called the principal-agent learning. Loizos [39] argues that when sensing its environment, an agent often receives

information that only partially describes the current state of affairs. The agent then attempts to predict what it has not sensed, by using other pieces of information available through its sensors. Machine learning techniques can naturally aid this task, by providing the agent with the rules to be used for making these predictions. For this to happen, however, learning algorithms need to be developed that can deal with missing information in the learning examples in a principled manner, and without the need for external supervision. It is shown that the Probably Approximately Correct semantics can be extended to deal with missing information during both the learning and the evaluation phase.

Numerous reinforcement learning applications have been recently reported in the literature. Some interesting examples include a proposal of reinforcement learning for agent-based production scheduling [63], a case-based reinforcement learning algorithm (CRL) for dynamic inventory control in a multi-agent supply-chain system [35]. Supervised learning techniques have been also applied to support agents learning capabilities. In [65], a support vector machine (SVM) based multiagent ensemble learning approach is proposed for credit risk evaluation. Different SVM learning paradigms with much dissimilarity are constructed as intelligent agents for credit risk evaluation. Multiple individual SVM agents are trained using training subsets. In the final stage, all individual results produced by multiple SVM agents in the previous stage are aggregated into an ensemble result.

3 Agent-Based Machine Learning

Recently, several machine learning solutions and techniques have been reported to rely on applying agent technologies. They belong to the two broad classes - universal one and dedicated to particular applications. Solutions and techniques belonging to the first class involve applications of the multi agent systems, including A-Teams and the population-based methods. This section contains a review of some recent universal and dedicated solutions with the exception of those based on the A-Team paradigm. Machine learning solutions using the A-Team paradigm are discussed in a detailed manner in Section 4.

3.1 Universal Solutions and Techniques

As it has been observed in [40] industry, science, and commerce fields often need to analyze very large datasets maintained over geographically distributed sites by using the computational power of distributed systems. The Grid can play a significant role in providing an effective computational infrastructure support for this kind of data mining. Similarly, the advent of multi-agent systems has brought us a new paradigm for the development of complex distributed applications. Through a combination of these two techniques an Agent Grid Intelligent Platform and an integrated toolkit VASstudio used as a testbed were proposed. Using grid platform as a testbed was also suggested in [50]. The author presents a parallel learning method for agents with an actor-critic architecture based on

artificial neural networks. The agents have multiple modules, where the modules can learn in parallel to further increase learning speed. Each module solves a sub-problem and receives its own separate reward signal with all modules trained concurrently. The method is used on a grid world navigation task showing that parallel learning can significantly reduce learning time.

Kitakoshi et al. [36] describe an on-line reinforcement learning system that adapts to environmental changes using a mixture of Bayesian networks. Machine learning approaches, such as those using reinforcement learning methods and stochastic models, have been used to acquire behavior appropriate to environments characterized by uncertainty. The results of several experiments demonstrated that an agent using the proposed system can flexibly adapt to various kinds of environmental changes.

Gifford in his Ph.D. dissertation [24] advocates an approach focused on the effects of sharing knowledge and collaboration of multiple heterogeneous, intelligent agents (hardware or software) which work together to learn a task. As each agent employs a different machine learning technique, the system consists of multiple knowledge sources and their respective heterogeneous knowledge representations. Experiments have been performed that vary the team composition in terms of machine learning algorithms and learning strategies employed by the agents. General findings from these experiments suggest that constructing a team of classifiers using a heterogeneous mixture of homogeneous teams is preferred.

Quteishat et.al. [49] proposed a neural network-based multi-agent classifier system using the trust, negotiation, and communication reasoning model. The main contribution of this work is that a novel trust measurement method, based on the recognition and rejection rates, was suggested.

Several important methods can be grouped under the umbrella of the collective or collaborative learning. In [27] it was shown how Evolutionary Dynamics (ED) can be used as a model for Qlearning in stochastic games. Analysis of the evolutionary stable strategies and attractors of the derived ED from the Reinforcement Learning (RL) application then predict the desired parameters for RL in Multi-Agent Systems (MASs) to achieve Nash equilibriums with high utility. Secondly, it was shown how the derived fine tuning of parameter settings from the ED can support application of the Collective Intelligence (COIN) framework. COIN is a proved engineering approach for learning of cooperative tasks in MASs. In [26] authors propose a collaborative machine learning framework to exploit inter-user similarities. More specifically, they present a kernel-based learning architecture that generalizes the well-known Support Vector Machine learning approach by enriching content descriptors with inter-user correlations.

Another umbrella covers learning classifier systems introduced by Holland [28] which use simple agents representing set of rules as a solution to a machine learning problem. A Pittsburgh-type LCS has a populations of separate rule sets, where the genetic algorithm recombines and reproduces the best of these rule sets. In a Michigan-style LCS there is only a single population and

the algorithm's action focuses on selecting the best classifiers within that rule set. Analysis of the properties of LCSs, comparison of several proposed variants and overview of the state of the art can be found in [4], [5], [64] and [9]. Useful extension of the LCS concept was proposed in [55]. This paper introduces a new variety of learning classifier system (LCS), called MILCS, which utilizes mutual information as fitness feedback. Unlike most LCSs, MILCS is specifically designed for supervised learning. Yet another extension introduces a mechanism for recognizing a current situation by determining a boundary between self and others, and investigates its capability through interaction with an agent [59]. An integration of several cognitively inspired anticipation and anticipatory learning mechanisms in an autonomous agent architecture, the Learning Intelligent Distribution Agent (LIDA) system was proposed in [44].

Ensemble techniques have proved to be very successful in boosting the performance of several types of machine learning methods. In [6] authors illustrate its usefulness in combination with GAssist, a Pittsburgh-style Learning Classifier System. Effective and competitive ensembles constructed from simple agents represented by expression trees induced using Gene Expression Programming have been proposed in [32] and [33]. Their approach has been tested using several ensemble constructing techniques including AdaBoost learning, voting pool of classifiers, incremental learning, cluster based learning, mass functions based learning and meta-learning.

Agent technology seems to be a natural tool for the distributed systems. Combining approaches to distributed learning with agent technology is considered as the promising and at the same time challenging problem in the distributed learning research [37]. In [67] an agent paradigm was proposed as a tool for integration of different techniques into an effective strategy of learning from data. The proposed hybrid learning system integrates basic components of the learning process. Data pre-processing, selection, transformation and induction of the learning and post-learning models are carried out by a set of agents cooperating during the task execution. Several agent-based architectures have already been proposed to solve the distributed learning problems. It is usually assumed that each site can have one or more associated agents, processing the local data and communicating the results to other agents that control and manage the knowledge discovery process. Examples include Papyrus [48], MALE [54], ANIMALS [56] and MALEF [61]. In [2] EMADS, a hybrid peer-to-peer agent based system comprising a collection of the collaborating agents distributed across a network, was described.

3.2 Dedicated Solutions and Techniques

In the machine learning literature numerous applications solving particular machine learning problem type or task where agent technology have played an important, even if supporting, role have been recently reported. In this short review the focus is on some example cases where agent technology has been used in an innovative manner.

Fan et.al. [23] have developed a two-stage model for personalized and intelligent information routing of online news. At the first stage, persistent user queries are extracted from rated documents based on Robertson's Selection Value (RSV). At the second stage, genetic programming is applied to discover the optimal ranking function for individual user. Pazzani and Billsus [46] developed a learning information agent called Syskill and Webert which could learn a user profile for the identification of interesting web documents. A separate user profile was created for each individual information topic. Web documents were represented as Boolean feature vectors, and each feature had a binary value indicating if a particular keyword appeared in the document or not. Feature selection was conducted based on Expected Information Gain which tends to select words appearing more frequently in positive documents. The classification mechanism of Syskill and Webert was based on a naive Bayesian classifier. The paper [3] focuses on symbolic transducers and recurrent neural preference machines to support the task of mining and classifying textual information. These encoding symbolic transducers and learning neural preference machines can be seen as independent agents, each one tackling the same task in a different manner.

Jansen [30] discusses various ways in which mobile agents could be applied to problem of detecting and responding to intrusions. Abraham et.al. [1] proposed a distributed Intrusion Detection System (IDS) consisting of several IDS over a large network, all of which communicate with each other, or with a central server that facilitates advanced network monitoring. In a distributed environment, system is implemented using co-operative intelligent agents distributed across the network. To detect intrusions in a network three fuzzy rule based classifiers are constructed. Moskovitch et.al. [43] conducted a comprehensive experiment for testing the feasibility of detecting unknown computer worms, employing several computer configurations, background applications, and user activities.

In [25], the authors utilize multi-agent machine learning and classifier combination to learn rock facies sequences from wireline well log data. The paper focuses on how to construct a successful set of classifiers, which periodically collaborate, to increase the classification accuracy. Utilizing multiple, heterogeneous collaborative learning agents is shown to be successful for this classification problem. Solving the pursuit problem with heterogeneous multiagent system using reinforcement learning was investigated in [29].

Zhang and Zhang [66] present , a multiagent data warehousing (MADWH) and multiagent data mining (MADM) approach for brain modeling. An algorithm named Neighbor-Miner is proposed for MADWH and MADM. The algorithm is defined in an evolving dynamic environment with semiautonomous neurofuzzy agents.

4 Machine Learning with A-Teams

Paradigms of the population-based methods and multiple agent systems have been during early nineties integrated within the concept of the asynchronous team of agents (A-Team). According to Talukdar et.al. [60] an asynchronous

team is a collection of software agents that cooperate to solve a problem by dynamically evolving a population of solutions. Current implementations of the A-Team concept are characterized by a high level of accessibility, scalability and portability. A review of the A-Team solutions and implementations can be found in [31]. All the machine learning solutions reviewed in this section are the A-Team implementations built using the JADE-Based A-Team middleware environment (JABAT) proposed in [7].

Two early applications of the A-Team paradigm to machine learning include training the cascade correlation learning architecture [15] and training the feed-forward artificial neural networks proposed in [16]. Recent solutions are focused on data reduction and distributed learning.

Data reduction in the supervised machine learning aims at deciding which instances and which features from the original training set should be retained for further use during the learning process. Data reduction is considered as an important step towards increasing effectiveness of the learning process when the available training sets are large or distributed and when the access to data is limited and costly. Data reduction performed without losing extractable information can result in increased capabilities and generalization properties of the learning model. It is obvious that removing some instances from the training set reduces time and memory complexity of the learning process. The data reduction algorithms can be divided into two categories: prototype selection and prototype extraction. Prototype selection is a technique of choosing a subset of reference vectors from the original set, also by reduction of attributes, whereas prototype extraction means the construction of an entirely new set of instances, smaller, in respect to its dimensionality, than the original dataset. Prototype extraction can also include the process of feature construction, where decreasing the number of attributes is carried-out by creating new features on the basis of some transformation of the original attributes. The performance criteria used in data reduction may include the accuracy of classification, the complexity of the hypothesis the classification costs and many other criteria.

An idea of applying agent technology to data reduction has been proposed in several papers of Czarnowski and Jedrzejowicz [14], [17], [18], [12], [20]. In the above papers several architectures, models and strategies for the A-Team based data reduction have been proposed. Using them usually improves quality of the respective supervised machine learning. Most competitive results have been obtained by A-Teams producing clusters of instances from the training set and then selecting instances from these clusters. Although a variety of methods could be used to produce clusters, using the similarity coefficient proposed in [14] and [17] as the clustering criterion have produced better than satisfactory results.

To solve the data reduction problem, several types of optimizing agents carrying out improvement procedures including tabu search, simulated annealing and variety of simple local search algorithms have been used. Basic assumptions behind the proposed approach are as follows:

- A solution is represented by a string consisting of two parts. The first contains numbers of instances selected as prototypes and the second – numbers of attributes chosen to represent the dataset.
- Prototype instances and attributes are selected from clusters through the population-based search carried out by the optimizing agents.
- Initially, potential solutions are generated by random selection of a single instance from each cluster and by random selection of the attribute numbers. Attributes are later adjusted by the attribute manager agent with a view to find the best combination and, at the same time, to unify the set of selected attributes at a global level (only in case of the distributed data reduction).

The solution manager is responsible for organizing the data reduction process through managing the population of solutions called individuals and updating them when appropriate. During the data reduction process the solution manager continues reading individuals (solutions) from the common memory and storing them back after attempted improvement until a stopping criterion is met. During this process the solution manager keeps sending randomly drawn individuals (solutions) from the common memory to optimizing agents. Each optimizing agent tries to improve the quality of the received solution and afterwards sends back the improved solution to the solution manager, which, in turn, updates common memory, replacing a randomly selected individual with the improved one. In each of the above cases the modified solution replaces the current one if it is evaluated as a better one. Evaluation of the solution is carried out by estimating classification accuracy of the classifier, which is constructed taking into account the instances and the attributes as indicated by the solution. Since the computational complexity of the above search procedures is linear, the computational complexity of the fitness evaluation is not greater than the complexity of the classifier induction. In case of the distributed data reduction an additional agent called attribute manager is used. Its role is to coordinate the attribute selection. The attribute manager agent is also responsible for the final integration of attributes selected locally by optimizing agents. The attribute manager actions include receiving candidate attributes from solution managers, and deciding on the common set of attributes to be used at both the local and the global levels.

An idea of applying A-Team paradigm to solving the distributed learning problem has been evolving since a couple of years. Different solutions were presented in several papers of Czarnowski and Jędrzejowicz [11], [19], [21], [22] and [13]. The proposed approach, denoted as LCDD (Learning Classifiers from Distributed Data), involves two stages, both supported by the collaboration between agents:

- Local, in which the selection of prototypes from the distributed data takes place (A-Teams are used to select prototypes by instance selection and/or removing irrelevant attributes).
- Global, consisting of pooling of the selected prototypes and producing the global learning model.

At the local level, that is, at the distributed data sources, agent-based population learning data reduction algorithms are executed in parallel. Instance and

attribute reduction are integrated with the classifier learning process. An important feature of the LCDD approach is A-Teams ability to select instances and attributes in cooperation between agents, thus assuring a homogenous set of prototypes at the global level. In this case, the instance selection is carried out independently at each site through applying the agent-based population search but the attribute selection is managed and coordinated through the process of interaction and collaboration between agents. All the required steps of the proposed approach are carried out by program agents of the four following types:

- Global level manager - agent responsible for managing the process of the distributed learning.
- Optimizing agent - agent executing a solution improvement algorithms.
- Solution manager - agent responsible for managing the population of solutions.
- Attribute manager - agent responsible for the attribute selection coordination.

The parallel process of data reduction at sites is managed by the global level manager. Its role is to manage all stages of the learning process. As the first step the global manager identifies the distributed learning task that is to be coordinated and allocates optimizing agents to the local sites using the available agent migration procedure. Then the global manager initializes parallel execution of all subtasks, that is data reduction processes at local sites. When all the subtasks have been completed, solutions from the local levels are used to produce the global solution. Producing it requires that the global manager is equipped with skills needed to induce the global classifier. When the prototypes obtained from local sites are homogenous then the local prototypes are integrated and the global manager creates the global classifier (meta-classifier), using some machine learning algorithm. When the local level solutions are represented by heterogeneous set of prototypes then the global classifier can be induced by applying one of the meta-classifier strategies like, for example, bagging, AdaBoost, majority voting or some hybrid strategy.

5 Conclusions

Main focus of this review is using agent technology in the field of machine learning with a particular interest on applying agent-based solutions to supervised learning. Some references are also made with respect to applying machine learning solutions to support agent technology. The review allows to formulate the following observations:

- Machine learning and agent technology are becoming more and more integrated bringing an important advantages to both fields.
- Machine learning can be seen as a prime supplier of learning capabilities for agent and multiagent systems.
- Agent technology have brought to machine learning several capabilities including parallel computation, scalability and interoperability.

- Agent-based solutions and techniques when applied to machine learning have proven to produce a synergetic effect coming from the collective intelligence of agents and a power of cooperative solutions generated through agent interactions.

Future research should help to further integrate both fields – agent technology and machine learning. Agent based solutions could be used to develop more flexible and adaptive machine learning tools. Collective computational intelligence techniques can be used to effectively solve computationally hard optimization and decision problems inherent to many supervised learning techniques and data reduction problems. Most promising direction for future research seems integration of machine learning and agent technology with a view to obtain effective solutions to the distributed learning problems. On the other hand more compact and reliable machine learning techniques are needed to equip agents with better learning capabilities.

References

1. Abraham, A., Jain, R., Thomas, J., Han, S.Y.: D-SCIDS: Distributed soft computing intrusion detection system. *J. Net.Comp. Appl.* 30, 81–98 (2007)
2. Albashiri, K.A., Coenen, F., Leng, P.: EMADS: An Extendible Multi-agent Data Miner. *Knowl. Bas. Syst.* 22, 523–528 (2009)
3. Arevian, G., Wermter, S., Panchev, C.: Symbolic state transducers and recurrent neural preference machines for text mining. *Int. J. Approx. Reason.* 32, 237–258 (2003)
4. Bacardit, J., Butz, M.V.: Data mining in learning classifier systems: comparing xcs with gassist. In: Kovacs, T., Llorà, X., Takadama, K., Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) *IWLCS 2003. LNCS (LNAI)*, vol. 4399, pp. 282–290. Springer, Heidelberg (2007)
5. Bacardit, J., Garrell, J.M.: Bloat Control and Generalization Pressure Using the Minimum Description Length Principle for a Pittsburgh Approach Learning Classifier System. In: Kovacs, T., Llorà, X., Takadama, K., Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) *IWLCS 2003. LNCS (LNAI)*, vol. 4399, pp. 59–79. Springer, Heidelberg (2007)
6. Bacardit, J., Krasnogor, N.: Empirical Evaluation of Ensemble Techniques for a Pittsburgh Learning Classifier System. In: Bacardit, J., Bernadó-Mansilla, E., Butz, M.V., Kovacs, T., Llorà, X., Takadama, K. (eds.) *IWLCS 2006 and IWLCS 2007. LNCS (LNAI)*, vol. 4998, pp. 255–268. Springer, Heidelberg (2008)
7. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: An Implementation of the JADE-base A-Team Environment. *Int. Trans.Syst. Sc.Appl.* 3(4), 319–328 (2008)
8. Boylu, F., Aytug, H., Koehler, G.J.: Principal-Agent Learning. *Dec. Supp. Syst.* 47, 75–81 (2009)
9. Bull, L., Kovacs, T.: Foundations of Learning Classifier Systems: An Introduction. *Stud. Fuzz. Soft Comp.* 183, 1–17 (2005)
10. Busni, L., Babuska, R., Schutter, B.: A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cyb.* 38, 156–171 (2008)
11. Czarnowski, I.: Distributed data reduction through agent collaboration. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009. LNCS*, vol. 5559, pp. 724–733. Springer, Heidelberg (2009)

12. Czarnowski, I.: Prototype Selection Algorithms for Distributed Learning. *Pat. Recogn.* 43, 2292–2300 (2010)
13. Czarnowski, I.: Distributed learning with data reduction. *LNCS Transactions on Collective Computational Intelligence IV*. Springer, Heidelberg (to appear, 2011)
14. Czarnowski, I., Jędrzejowicz, P.: An Approach to Instance Reduction in Supervised Learning. In: Coenen, F., Preece, A., Macintosh, A. (eds.) *Research and Development in Intelligent Systems XX*, pp. 267–282. Springer, London (2004)
15. Czarnowski, I., Jędrzejowicz, P.: An Agent-Based PLA for the Cascade Correlation Learning Architecture. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) *ICANN 2005*. LNCS, vol. 3697, pp. 197–202. Springer, Heidelberg (2005)
16. Czarnowski, I., Jędrzejowicz, P.: An Agent-based Approach to ANN Training. *Knowl.-Based Syst.* 19, 304–308 (2006)
17. Czarnowski, I., Jędrzejowicz, P.: An Agent-Based Approach to the Multiple-Objective Selection of Reference Vectors. In: Perner, P. (ed.) *MLDM 2007*. LNCS (LNAI), vol. 4571, pp. 117–130. Springer, Heidelberg (2007)
18. Czarnowski, I., Jędrzejowicz, P.: An agent-based algorithm for data Reduction. In: Bramer, M., Coenen, F., Petridis, M. (eds.) *Research and Development of Intelligent Systems XXIV*, pp. 351–356. Springer, London (2007)
19. Czarnowski, I., Jędrzejowicz, P.: A Comparison Study of Strategies for Combining Classifiers from Distributed Data Sources. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) *ICANNGA 2009*. LNCS, vol. 5495, pp. 609–618. Springer, Heidelberg (2009)
20. Czarnowski, I., Jędrzejowicz, P.: An Approach to Data Reduction and Integrated Machine Classification. *New Gen. Comp.* 28, 21–40 (2010)
21. Czarnowski, I., Jędrzejowicz, P.: An agent-based framework for distributed Learning. *Eng. Appl. Art. Intel.* 24, 93–102 (2011)
22. Czarnowski, I., Jędrzejowicz, P., Wierzbowska, I.: An A-Team Approach to Learning Classifiers from Distributed Data Sources. *Int. J. Intel. Inf. Db. Syst.* 4(3), 245–263 (2010)
23. Fan, W., Gordon, M., Pathak, P.: An integrated two-stage model for intelligent information routing. *Dec. Sup. Syst.* 42(1), 362–374 (2006)
24. Gifford, C.: M.: *Collective Machine Learning: Team Learning and Classification in Multi-Agent Systems*. Ph.D. dissertation, University of Kansas (2009)
25. Gifford, C.M., Agah, A.: Collaborative multi-agent rock facies classification from wireline well log data. *Eng. Appl. Art. Intel.* 23, 1158–1172 (2010)
26. Hofmann, T., Basilico, J.: Collaborative Machine Learning. In: Hemmje, M., Niederée, C., Risse, T. (eds.) *From Integrated Publication and Information Systems to Information and Knowledge Environments*. LNCS, vol. 3379, pp. 173–182. Springer, Heidelberg (2005)
27. Hoenl, P.J., Tuyls, K.: Analyzing Multi-agent Reinforcement Learning Using Evolutionary Dynamics. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *ECML 2004*. LNCS (LNAI), vol. 3201, pp. 168–179. Springer, Heidelberg (2004)
28. Holland, J.H.: Escaping Brittleness: The possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) *Machine Learning, an Artificial Intelligence Approach*, vol. II, pp. 593–623. Morgan Kaufmann, Palo Alto (1986)
29. Ishiwaka, Y., Sato, T., Kakazu, Y.: An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Rob. Autonom. Syst.* 43, 245–256 (2003)

30. Jansen, W.A.: Intrusion detection with mobile agents. *Comp. Comm.* 25, 1392–1401 (2002)
31. Jędrzejowicz, P.: A-Teams and Their Applications. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS*, vol. 5796, pp. 36–50. Springer, Heidelberg (2009)
32. Jędrzejowicz, J., Jędrzejowicz, P.: A Family of GEP-Induced Ensemble Classifiers. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS*, vol. 5796, pp. 641–652. Springer, Heidelberg (2009)
33. Jędrzejowicz, J., Jędrzejowicz, P.: Two Ensemble Classifiers Constructed from GEP-Induced Expression Trees. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010. LNCS*, vol. 6071, pp. 200–209. Springer, Heidelberg (2010)
34. Jennings, N., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. *Aut. Ag. Multi-Ag. Syst.* 1, 7–38 (1998)
35. Jiang, C., Sheng, Z.: Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system. *Exp. Syst. Appl.* 36, 6520–6526 (2009)
36. Kitakoshi, D., Shioya, H., Nakano, R.: Empirical analysis of an on-line adaptive system using a mixture of Bayesian networks. *Inf. Sc.* 180, 2856–2874 (2010)
37. Klusch, M., Lodi, S., Moro, G.L.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) *Intelligent Information Agents. LNCS (LNAI)*, vol. 2586, pp. 104–122. Springer, Heidelberg (2003)
38. Liau, C.J.: Belief, information acquisition, and trust in multi-agent systems - A modal logic formulation. *Artif. Int.* 149(1), 31–60 (2003)
39. Loizos, M.: Partial observability and learnability. *Artif. Int.* 174, 639–669 (2010)
40. Luo, J., Wang, M., Hu, J., Shi, Z.: Distributed data mining on Agent Grid: Issues, platform and development toolkit. *Fut. Gen. Comp. Syst.* 23, 61–68 (2007)
41. Mannor, S., Shamma, J.S.: Multi-agent learning for engineers. *Artif. Int.* 171, 417–422 (2007)
42. Masoumi, B., Meybodi, M.R.: Speeding up learning automata based multi agent systems using the concepts of stigmergy and entropy. *Exp. Syst. Appl.* (to appear, 2011)
43. Moskovitch, R., Elovici, Y., Rokach, L.: Detection of unknown computer worms based on behavioral classification of the host. *Comp. Stat. Data Anal.* 52, 4544–4566 (2008)
44. Negatu, A., D’Mello, S.K., Franklin, S.: Cognitively Inspired Anticipatory Adaptation and Associated Learning Mechanisms for Autonomous Agents. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) *ABiALS 2006. LNCS (LNAI)*, vol. 4520, pp. 108–127. Springer, Heidelberg (2007)
45. Nowé, A., Verbeeck, K., Peeters, M.: Learning automata as a basis for multi agent reinforcement learning. In: Tuyls, K., Hoen, P.J., Verbeeck, K., Sen, S. (eds.) *LAMAS 2005. LNCS (LNAI)*, vol. 3898, pp. 71–85. Springer, Heidelberg (2006)
46. Pazzani, M., Billsus, D.: Learning and revising user profiles: the identification of interesting web sites. *Mach. Learn.* 27(3), 313–331 (1997)
47. Preux, P., Delepouille, S., Darcheville, J.-C.: A generic architecture for adaptive agents based on reinforcement learning. *Inf. Sc.* 161, 37–55 (2004)
48. Prodromidis, A., Chan, P.K., Stolfos, S.J.: Meta-learning in Distributed Data Mining Systems: Issues and Approaches. In: Kargupta, H., Chan, P. (eds.) *Advances in Distributed and Parallel Knowledge Discovery*, vol. 3, AAAI/MIT Press, Menlo Park (2000)

49. Quteishat, A., Lim, C.P., Tweedale, J., Jain, L.C.: A neural network-based multi-agent classifier system. *Neurocomp.* 72, 1639–1647 (2009)
50. Raicevic, P.: Parallel reinforcement learning using multiple reward signals. *Neurocomp.* 69, 2171–2179 (2006)
51. Rosaci, D.: CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Inf. Sys.* 32, 793–825 (2007)
52. Sardinha, J.A.R.P., Garcia, A., de Lucena, C.J.P., Milidiú, R.L.: A Systematic Approach for Including Machine Learning in Multi-agent Systems. In: Bresciani, P., Giorgini, P., Henderson-Sellers, B., Low, G., Winikoff, M. (eds.) AOIS 2004. LNCS (LNAI), vol. 3508, pp. 198–211. Springer, Heidelberg (2005)
53. Shoham, Y., Powers, R., Grenager, T.: If multi-agent learning is the answer, what is the question? *Artif. Int.* 171(7), 365–377 (2007)
54. Sian, S.: Extending Learning to Multiple Agents: Issues and a Model for Multi-Agent Machine Learning (Ma-MI). In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 440–456. Springer, Heidelberg (1991)
55. Smith, R.E., Jiang, M.K., Bacardit, J., Stout, M., Krasnogor, N., Hirst, J.D.: A learning classifier system with mutual-information-based fitness. *Evol. Int.* 1(3), 31–50 (2010)
56. Stolfo, S., Prodromidis, A.L., Tselepis, S., Lee, W., Fan, D.W.: JAM: Java Agents for Meta-learning over Distributed Databases. In: 3rd International Conference on Knowledge Discovery and Data Mining, pp. 74–81. AAAI Press, Newport Beach (1997)
57. Sutton, R.S., Barto, A.G.: Reinforcement Learning. An Introduction. MIT Press, Cambridge (1998)
58. Symeonidis, A.L., Chatzidimitriou, K.C., Athanasiadis, I.N., Mitkas, P.A.: Data mining for agent reasoning: A synergy for training intelligent agents. *Eng. Appl. Artif. Int.* 20, 1097–1111 (2007)
59. Takadama, K., Inoue, H., Shimohara, K., Okada, M., Katai, O.: Agent architecture based on an interactive self-reflection classifier system. *Artif. Life Rob.* 5, 103–108 (2001)
60. Talukdar, S., Baerentzen, L., Gove, A., De Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous Agents. *J. Heur.* 4(4), 295–321 (1998)
61. Tozicka, J., Rovatsos, M., Pechoucek, M., Urban, U.: MALEF: Framework for Distributed Machine Learning and Data Mining. *Int. J. Int. Inf. Db. Sys.* 2(1), 6–24 (2008)
62. Tweedale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., Phillips-Wren, G.E.: Innovations in multi-agent systems. *J. Net. Comp. Appl.* 30(3), 1089–1115 (2007)
63. Wang, Y.-C., Usher, J.M.: Application of reinforcement learning for agent-based production scheduling. *Eng. Appl. Artif. Int.* 18, 73–82 (2005)
64. Wilson, S.W.: State of XCS Classifier System Research. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) IWLCS 1999. LNCS (LNAI), vol. 1813, pp. 63–81. Springer, Heidelberg (2000)
65. Yu, L., Yue, W., Wang, S., Lai, K.K.: Support vector machine based multiagent ensemble learning for credit risk evaluation. *Exp. Sys. Appl.* 37, 1351–1360 (2010)
66. Zhang, W.-R., Zhang, L.: A multiagent data warehousing (MADWH) and multi-agent data mining (MADM) approach to brain modeling and neurofuzzy control. *Inf. Sc.* 167, 109–127 (2004)
67. Zhang, S., Wu, X., Zhang, C.: Multi-Database Mining. *IEEE Computational Intelligence Bulletin* 2(1), 5–13 (2003)

Goal Orientated Conversational Agents: Applications to Benefit Society

Keeley Crockett, James O'Shea, and Zuhair Bandar

The Intelligent Systems Group, School of Computing, Mathematics & Digital Technology,
Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK
{K.Crockett, J.D.OShea, Z.Bandar}@mmu.ac.uk

Abstract. Goal-orientated Conversational Agents are a specific family of conversational agents that are designed to converse with humans through the use of natural language dialogue to achieve a specific task. Traditionally, they utilise pattern matching algorithms to capture the values of specific attributes through their values through dialogue interaction with a user. This is achieved through the use of scripts which contain sets of rules about the domain and a knowledge base to guide the conversation towards achieving a specific goal. Such systems are ideal for providing clear and consistent advice 24 hours a day in many different scenarios, including advising employees about their organisations policies and procedures, guiding a user through buying a suitable product, and tutoring a student to understand a learning objective. This paper presents an overview of a methodology for constructing goal orientated conversational agents. Three case studies which employ this methodology are introduced and evaluated.

Keywords: Goal-orientated conversational agents, pattern matching.

1 Introduction

A conversational agent (CA) is an agent which uses natural language dialogue to communicate with users [1]. The CA will also have the ability to reason and pursue a course of action based on its interactions with humans and other agents. An automated and interactive conversational agent system should be able to provide anonymous and 24-hour access to humans in a variety of applications e.g. customer self service on the web, internal advice on an organisation's policies and procedures, and would allow users to be able to ask questions in natural language. At the same time, the information / advice given by the CA would always be consistent, appropriate and valid whilst the agent can be designed to exhibit sympathetic or compassionate behaviour to a particular circumstance. A further strength is that CA's can be tailored to behave in a way that reflects an organization's or businesses culture and to have distinctive personalities.

A Goal-Oriented CA (GO-CA) is a type of conversational agent which has a deep strategic purpose which enables it to direct a conversation to achieve a goal [2]. Thus the GO-CA [2, 3] may spend more time leading the conversation and asking questions

than the human participant which makes it different from systems such as chatterbots and traditional conversational agents. Chatterbots [4] attempt to prolong social conversational with a human for as long as possible generally using pointless chat [2]. Traditional conversational agents either focus on the embodied component (e.g. graphical representation of the agents face) with simple dialogue or are text based and require complex scripting procedures to appear to have intelligence in answering the user's enquiry. This paper first introduces a brief history of conversational agents and in section 3 a methodology for constructing GO-CA's is introduced. Sections 4, 5 and 6 describe three diverse applications of GO-CA which illustrate the diverse application of the GO-CA methodology. Finally section 7 highlights some of the main issues in developing CA's and areas of future research.

2 Milestones in Conversational Agent Development

The best known early CA was Eliza [6]. ELIZA's main trick was to use questions to draw a conversation out of the user. However the main criticism of ELIZA was the program's lack of an internal world model that could influence and track conversation [6]. ELIZA was followed by PARRY [7] which simulated paranoid behaviour effectively in a blind, Turing like, test by doctors. This took advantage of the fact that unusual responses from PARRY would be attributed to paranoia. PARRY was followed by a number of CA's which utilised pattern matching techniques based on Infochat [8] which implemented an interpreter for a language known as Pattern Script. The most recent innovation in pattern matching is symbolic reduction which is used in A.L.I.C.E. [9]. Symbolic reduction allows a rule to remove part of the user utterance and re-submit the remainder to the CA for further analysis.

A significant proportion of CA research has been dedicated towards embodied agents where expressions and gestures contribute as much as the natural language dialogue [10]. The purpose is to make a computer application appear more human like and become more engaging with the user. While substantial amounts of work [11, 12] have gone into evaluating such interfaces in terms of features such as visual appearance, expressiveness, personality, presence, role and initiative, less attention has been applied to the evaluation of the actual conversation.

CA's are also being used to allow non-programmers to query databases [13,14]. Owda et al [13] proposes the use of goal oriented conversation to provide the natural language interface and helps disambiguate the user's queries, by utilizing dialogue interaction. CA's have been successfully used within Conversational Intelligent Tutoring Systems (CITS) [15]. CITS are computer-based educational systems which employ intelligent technologies to provide individualised instruction in natural language and help students by providing personalised tutoring at a time and a pace to suit the individual. Case study three will describe CITS in more detail. A number of companies have developed CA's to either sell products or provide customer support. VirtuOz, use conversational agents for customer support such as in online sales, advice and recommendations [16].

3 Constructing Goal Orientated Conversational Agents (GO-CA)

This section describes an architecture developed by the Intelligent Systems Group at MMU for constructing GO-CAs [2]. The methodology is designed so the user can approach the GO-CA with a problem or a request for information which can then lead to interactive dialogue between the user and the GO-CA until the user's goal is achieved. This is achieved in current implementations using a rule-based system, which contains a model of the problem domain that is expressed in terms of a set of attributes. Through the process of dialogue, appropriate attributes are captured (through pattern matching algorithms) to model the particular problem experienced by the user and identify the appropriate solution. A further strength of a GO-CA is that it will engage in extended dialogue, during the course of which it will appear to have mental states that are directed at states of affairs in the world e.g. beliefs, desires, hopes, perception, intention etc [2]. The GO-CA is a mixed-initiative system (from time to time either the human or the agent may take control of the conversation) [2]. However, due to the nature of the GO-CA, if the human diverts the conversation from the goal, the GO-CA will always attempt to get the conversational back on track using the rule base and knowledge obtained from attribute values captured during the conversation. Figure 1 shows the generic modular architecture for a typical GO-CA [3]. This is intended to take on challenging real-world applications in which the user may present adversarial, disruptive or deceptive behavior at times during the conversation [2].

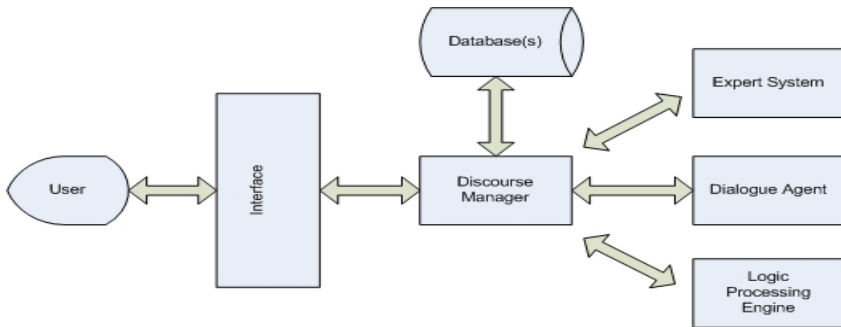


Fig. 1. GO-CA Architecture

Each component within the architecture will now be briefly described:

- The *discourse manager* is the central component of the GO-CA and provides communication with the problem specific expert system (rule base), dynamic graphical user interface (GUI) and the Dialogue Agent (DA). When events take place in the GUI e.g. the user enters dialogue, the discourse manager requests a response from the expert system or the DA, and then instructs the GUI to update its display accordingly.
- The *expert system* contains a rule base comprising of the domain knowledge. Through dialogue with the DA, the rule base will gather information about the user's circumstances and will inform the discourse manager which attributes and its associated value needs to be captured in order to progress.

- The *graphical user interface* component manages the display and responds to the users requests either in the form of mouse click, button selection or natural language dialogue.
- The *Dialogue Agent* undertakes dialogue with the user and has the ability to understand natural language queries, and formulate a response. The role of the DA is to capture information in the form of attributes to answer questions about the specific domain. The DA obtains its own dialogue from a number of scripts, each representing a given context. Each script contains a series of rules and patterns which have been semi-automatically scripted by humans who have gained knowledge of the domain through the knowledge engineering process. An example rule for dealing with an employee who expresses confusion in natural language about the question being asked by the GO-CA is shown below:

```

<Rule_04>
p:50 *<confused-0>*
p:50 *<confusing-0>*
p:50 *<sure-neg-0>*
p:50 *<sure-neg-1>*
p:50 *help*
p:50 *not *<understand-0>*
r: How can I help you

```

where p is the strength of the pattern and r the response. Patterns can contain wild-card elements “*” which will match with one or more consecutive characters. In addition, the macro “<confused-0>” enables the scripter to incorporate stock patterns into a rule [9].

The construction of each GO-CA comprises of the following high level stages:

- Knowledge Engineering – Capturing and structuring of knowledge about the specific problem domain and the organisations culture and beliefs.
- Design and development of a rule based expert system to achieve user goals.
- User Interface Design (Web/ Intranet/ mobile / PDA/ voice recognition).
- Dialogue script design and organisation.
- Internal database integration for data validation or information extraction.
- Custom-built architecture involving the integration with the organisations existing systems.

4 Case Study 1: Student Debt Advisor

The first case study describes a GO-CA called Adam which was developed by Convagent Ltd [17]. Adam is a UK University Student Debt Advisor and is highly focused on providing advice to students by finding out why they have not paid their debts and offering advice on how they can find ways to pay them. The nature of support offered has required Adam to cope with upset and angry students, and deal with abusive language. Student debt is a growing concern in the UK with the average cost of a three-year University degree set to range from £38000 upwards [18]. ‘Adam’

was designed to stimulate the behaviour of student debt advisors within the university and give clear and consistent advice on what help was available to students who had debt problems.

The GO-CA was designed to capture all the relevant information from the student's natural language dialogue in order to help them with their debt. Knowledge trees were used to represent the policies for student debt management within the organization. The GO-CA, guided by the knowledge tree had awareness of the type of attributes that could be provided by the student, such as database attribute names, and values, dates, and comparative terms. In order to achieve this, a different script was developed to identify the presence or absence of each possible relevant attribute and its value that would allow the GO-CA to guide the user through the University student debt advisor process. Example dialog between a student (Tom) and Adam can be seen in Figure 2.

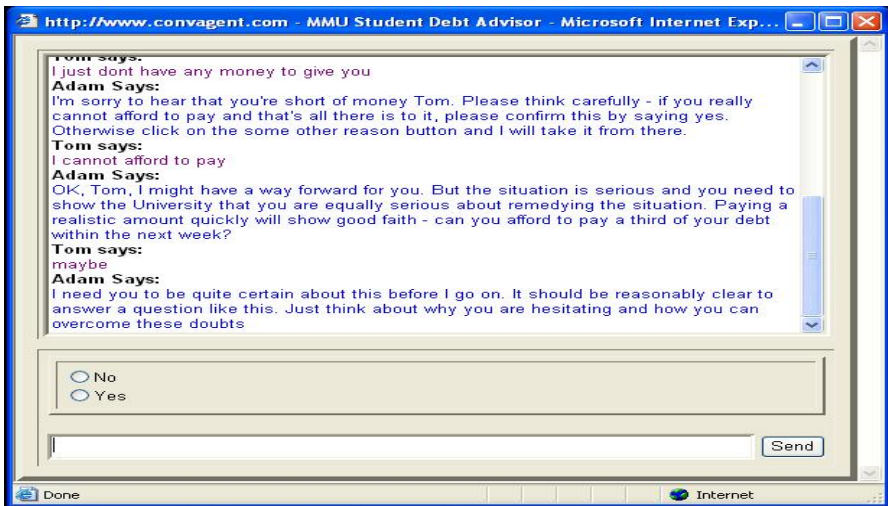


Fig. 2. Sample interaction with 'Adam'

4.2 Evaluation

The GO-CA, Adam was subjected to usability testing with 200 undergraduate students before going live. Students were given one of 5 scenarios which were designed from real life student circumstances of debt. For example, one such scenario is about a student called Tom who has got behind with his fee payments, has sent a late cheque and in the mean time has received a warning letter from the University. This scenario was selected as it is one of the two most common reasons for students to call the University human helpline.

Students were asked to provide feedback of their experience of 'Adam', using a questionnaire. 80% of students were happy with the speed in which Adam elicited the responses; 70% of students were satisfied by the advice provided by Adam and 75%

thought that Adam answered any additional queries that they had satisfactory. In the last part of the questionnaire, students were asked, "If you needed advice about your student debt problems, where would you go first?" 47% of students stated that they would use Adam instead of visiting the University Finance Office while 20% stated they would use Adam instead of telephoning the University Finance Office. From this initial testing phase, the majority of student's comments were positive. For example "He needs to be a bit friendly, I felt like I was seeing a real Advisor.... No wait that's a good thing!" However, the testing phase identified a number of areas where Adam could be improved such as, a better understanding of student life experiences and the use of slang and mobile text talk. These ideas were then incorporated into the 'live' version of Adam which is currently being run at the University. Further studies are on-going and feedback obtained from the GO-CA log files enables 'Adam' to continue to learn and relate to life as a student.

5 Case Study 2: HR Bullying and Harassment Advisor

The second case study will describe how a GO-CA was used to act as an advisor for employees on 'Bullying and Harassment in the Workplace' policy and procedures in a large organisation [19]. In the UK, no single piece of legislation addresses the problem of bullying and harassment [20]. Rather, organisations are required to implement a number of different laws which protect employees from harassment due to various causes. This can result in the development of complex policy documents. An individual may not be able to apply the policy or procedure to their personal situation. This type of policy will often require additional training and guidance to support for members of the organisation wishing to understand the reporting problem. The high support cost and fairly static nature of such policies means that they are suitable for automation using a CA. This would allow anonymous, consistent and appropriate advice to be available 24/7. Knowledge engineering was used to elicit the main questions in relation to bullying and harassment asked by employees within the organisation [19]. Then a rule base was then used to structure the bullying and harassment domain. Figure 3 shows a portion of the rule base for structuring knowledge about the 'I am being victimized' option. Through natural language dialogue with the user, the GO-CA looks for each possible relevant attribute and any associated value in the user's input in order to determine what rule in the rule base to fire and hence what advice to give.

Figure 4 shows an example of a dialogue interaction between an employee and the bullying and harassment advisor 'Adam'. The dynamic interface presents the user with three methods of communication: graphical or through natural language dialogue or in a mixed mode. The interface was designed so that more experienced users or those with disabilities were able to select options by simply clicking on their choices. Alternatively, natural language dialogue can also be entered in order to discuss a person's individual circumstances, explain concepts and offer advice at any stage in the process.

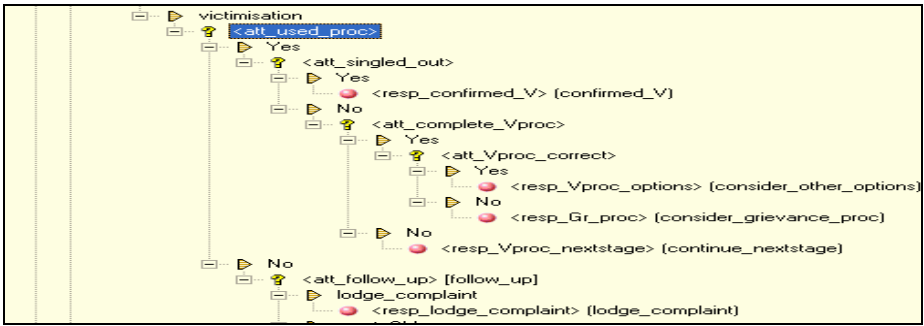


Fig. 3. Portion of rule base

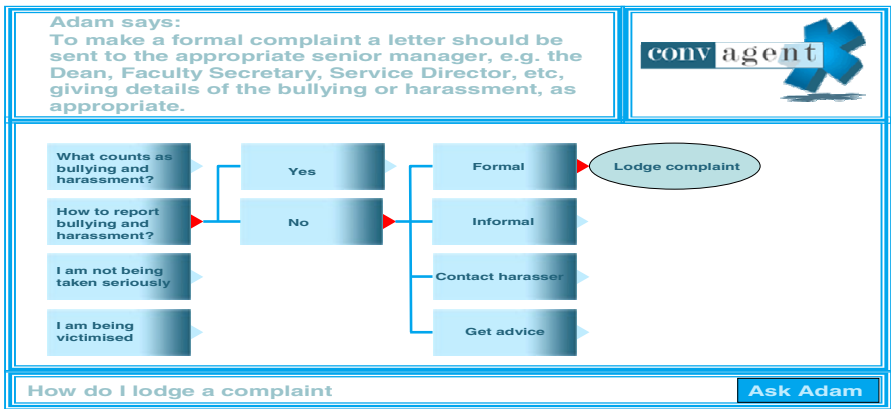


Fig. 4. Bullying and Harassment Advisor

5.2 Evaluation

A representative group of 30 employees' were selected to evaluate the GO-CA. The group of 30 employees worked in different roles and for diverse companies, aged between 25 and 45. The group included both male and female members from administrative, academic and managerial backgrounds. A scenario of possible sexual harassment was developed along with a questionnaire to record feedback. The group were asked to read the scenario and then complete two tasks within the system. Users then completed an electronic evaluation questionnaire anonymously. Users felt that the system was intuitive and easy to use, giving high scores to ease of navigation and the ability to find information. On the whole, scores were high and the group found the system understandable and intuitive to use. 94% of users indicated that they had found the advice they sought without difficulty and one user commented that they "did not need to 'learn' to use the advisor". This is a very important benefit as users who seek to use this type of system do so because they want the same level of advice as provided by a human, without being constrained due to their technical ability.

6 Case Study 3 Conversational Intelligent Tutoring System

Case study 3, describes a Conversational Intelligent Tutoring System (CITS) known as Oscar which can dynamically predict and adapt to a learner's learning style during the tutoring session [21]. The system's goal is for a student to complete a tutorial which dynamically adapts to their learning style. Oscar CITS imitates a human tutor by leading a tutorial in natural language, intelligently analysing solutions and offering problem solving support rather than simply presenting the answers. In providing the learner with the most appropriate learning material for their learning style, Oscar CITS aims to improve the effectiveness of the learning experience and provoke a deeper understanding of the topic, and thus improve confidence. Learning styles are central to the Oscar CITS, so development started by considering the *Index of Learning Styles* (ILS) model [22] which was developed to describe the learning styles in engineering education and suggests different pedagogical styles to address learners' needs. Following an initial study, a subset of the best ILS predictor questions for each learning style dimension was then considered during the development of the Oscar CITS. The domain of SQL was selected due to its wide use in undergraduate courses. Tutoring revision scenarios were designed based around the syllabus and the database lecturers' experience of revision tutorials. Each revision question was mapped to the ILS model using the model's descriptions of indicative behaviour, such as a preference for theoretical questions. Following on from the tutorial design, the dialogue for the tutorial was scripted using Convagent's GO-CA [17]. Overall, there were 38 contexts containing around 400 rules which demonstrates the complexity of developing a CITS. The full architecture and methodology for developing Oscar CITS is described in [21]. Throughout tutoring, the CITS logs information including the behaviour of the participant, their knowledge of the topic and the adaptation employed.

6.1 Evaluation of Oscar

An initial study comprising of 35 native English speakers was conducted to investigate the adaptation of tutoring to an individual's learning styles for three ILS dimensions. Each participant had had previous experience of an undergraduate SQL course. Initially, participants were asked to complete the formal ILS questionnaire followed by a multiple choice test to assess existing SQL knowledge. Next, they engaged in a personalised tutoring conversation led by Oscar. During tutoring, each participant answered questions and completed various tasks in SQL. Depending on their level of knowledge, participants were exposed to various resources and given hints and help as required. Finally, each participant was asked to complete the same multiple choice test to measure their learning over the session. The results of the gave an accuracy of predicting learning style on the Sequential/Global dimension of 75-80%. Overall, the results have shown that the Oscar CITS tutoring seemed to help learning as all participants who did not initially achieve full marks improved their test scores by an average of 25%.

7 Conclusions and Future Research Directions

User evaluation of each of the 3 diverse GO-CA's has been positive and has illustrated that such an agent can help a user achieve a sophisticated goal through natural language dialogue. Whilst the three cases described in this paper highlight the clear benefits of using GO-CA, there are a number of research opportunities to improve the methodology. Firstly, capturing the domain knowledge and then scripting a GO-CA is a labour-intensive process. Changes in the environment and/or in the knowledge would effectively mean that new scripts would need to be created meaning the maintenance of such systems is high and the process of incorporating new scripts could lead to conflicting rules in the rule base. An alternative approach proposed by O'Shea et al [23] replaces the pattern matching approach to scripting by instead making use of sentence similarity measures. Sentence Similarity Based on Semantic Nets and Corpus Statistics [24...26] is a measure that focuses directly on computing the similarity between very short texts of sentence length. In the proposed methodology, sentence similarity measures [24...26], are employed to examine semantics rather than pattern matching. Initial results have indicated a substantial reduction in the GO-CA's rule base as scripted patterns in all rules are instead replaced with one or two natural language sentences. Secondly, evaluating any form of text based conversational agents is difficult to achieve as there is no clear scientific benchmark. Early evaluation methodologies such as PARADISE [27] were designed for speech-based conversational agents where the overall goal in the conversation was user satisfaction. A more scientific approach has been developed by O'Shea et al [29] who has developed a benchmark data set of 65 sentence pairs with human-derived similarity ratings. This data set will be a viable measure in the evaluation of *all* text based CA's in the future.

References

1. Massaro, D.W., Cohen, M.M., Beskow, J., Cole, R.A.: Developing and evaluating conversational agents. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (eds.) *Embodied Conversational Agents*. MIT Press, Cambridge (2000)
2. O'Shea, J., Bandar, Z., Crockett, K.: Systems Engineering and Conversational Agents. In: *Intelligent-Based Systems Engineering*. ISRL, vol. 10, pp. 210–232. Springer, Heidelberg (2011)
3. Crockett, K., Bandar, Z., O'Shea, J., Mclean, D.: Goal Orientated Conversational Agents - the rocky road to commercialization. In: *IEEE WCCI, Barcelona*, pp. 1820–1827 (2010)
4. Mauldin, M.: *Chatterbots, Tinymuds, And The Turing Test*. AAAI, Menlo Park (1994)
5. Turing, A.M.: Computing Machinery and Intelligence. *Mind* 59(236), 433–460 (1950)
6. Weizenbaum, J.: ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the Ass' for Computing Machinery* 9, 36–45 (1966)
7. Colby, K.: *Artificial Paranoia: A Computer Simulation of Paranoid Process*. Pergamon Press, Oxford (1975)
8. A.L.I.C.E. AI Foundation, Inc. 2011, The A.L.I.C.E. Artificial Intelligence Foundation (2011), <http://alice.pandorabots.com/> (viewed January 13, 2011)
9. Michie, D., Sammut, C.: *Infochat Scripter's Manual*. Convagent Ltd., Manchester (2001)

10. Cassell, J., Sullivan, J., Prevost, S., Churchill, E.: *Embodied Conversational Agents*. MIT press, Cambridge (2000)
11. Cassell, J., Gill, A., Tepper, P.: Coordination in Conversation and Rapport. In: *Proceedings of the Workshop on Embodied Natural Language*, Association for Computational Linguistics (2007)
12. Massaro, D.: A framework for evaluating multimodal integration by humans and a role for embodied conversational agents, pp. 24–31 (2004)
13. Owda, M., Crockett, K., Bandar, Z.: Conversation-Based Natural Language Interface to Relational Database. In: *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 363–367 (2007)
14. Pudner, K., Crockett, K., Bandar, Z.: An Intelligent Conversational Agent Approach to Extracting Queries from Natural Language, World Congress on Engineering. In: *International Conference of Data Mining and Knowledge Engineering*, pp. 305–310 (2007)
15. Cha, H.J., Kim, Y.S., Park, S.-H., Yoon, T.-b., Jung, Y.M., Lee, J.-H.: Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 513–524. Springer, Heidelberg (2006)
16. VirtuOz (2011), <http://www.virtuoz.com> (accessed January 13, 2011)
17. Convagent Ltd. (2011), <http://www.convagent.com> (accessed January 13, 2011)
18. University funding (2011), <http://www.bbc.co.uk/news/education-11483638> (accessed January 12, 2011)
19. Latham, A., Crockett, K., Bandar, Z.: A Conversational Expert System Supporting Bullying And Harassment Policies. In: *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence*, pp. 163–168 (2010)
20. *Bullying in the workplace* (2011), [http://www.direct.gov.uk/en/Employment/Employees/DiscriminationAtWork/DG_10026670](http://www.direct.gov.uk/en/Employment/Employees/Employees/DiscriminationAtWork/DG_10026670) (accessed January 13, 2011)
21. Latham, A., Crockett, K., McLean, D., Edmonds, B.: Predicting Learning Styles in a Conversational Intelligent Tutoring System. In: Luo, X., Spaniol, M., Wang, L., Li, Q., Nejdil, W., Zhang, W. (eds.) *ICWL 2010*. LNCS, vol. 6483, pp. 131–140. Springer, Heidelberg (2010)
22. Felder, R., Silverman, L.K.: Learning and Teaching Styles in Engineering Education. *J. Engineering Education* 78(7), 674–681 (1988)
23. O’Shea, K., Bandar, Z., Crockett, K.: Towards a New Generation of Conversational Agents Based on Sentence Similarity, *Lecture Notes in Engineering*, In: *Advances in Electrical Engineering and Computational Science*, vol. 39, pp. 1100–1876. Springer, Heidelberg (2009)
24. Li, Y., Bandar, Z., Mclean, D.: Measuring semantic similarity between words using lexical knowledge and neural networks. In: Yin, H., Allinson, N.M., Freeman, R., Keane, J.A., Hubbard, S. (eds.) *IDEAL 2002*. LNCS, vol. 2412, pp. 111–116. Springer, Heidelberg (2002)
25. Li, Y., Bandar, Z., O’Shea, J., Mclean, D., Crockett, K.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering* 18(8), 1138–1150 (2006)
26. Li, Y., Bandar, Z., McLean, D.: An approach for Measuring Semantic Similarity Between Words Using Multiple Information Sources. *IEEE Transactions on Knowledge and Data Engineering* 15(4), 871–881 (2003)
27. Sanders, G.A., Scholtz, J.: Measurement and Evaluation of Embodied Conversational Agents. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (eds.) *Embodied Conversational Agents* (2000)
28. O’Shea, J., Bandar, Z., Crockett, K., Mclean, D.: Benchmarking short text semantic similarity. *Int. J. Intelligent Information and Database Systems* 4(2) (2010)

Using a Slim Function Word Classifier to Recognise Instruction Dialogue Acts

James O'Shea, Zuhair Bandar, and Keeley Crockett

School of Computing, Mathematics & Digital Technology,
Manchester Metropolitan University, Chester St., Manchester M1 5GD, United Kingdom
{j.d.oshea, z.bandar, k.crockett}@mmu.ac.uk

Abstract. This paper extends a novel technique for the classification of short texts as Dialogue Acts, based on structural information contained in function words. It investigates the new challenge of discriminating between instructions and a non-instruction mix of questions and statements. The proposed technique extracts features by replacing function words with numeric tokens and replacing each content word with a standard numeric wildcard token. Consequently this is a potentially challenging task for the function-word based approach as the salient feature of an instruction is an imperative verb, which will always be replaced by a wildcard. Nevertheless, the results of the decision tree classifiers produced provide evidence for potentially highly effective classification and they are comparable with initial work on question classification. Improved classification accuracy is expected in future through optimisation of feature extraction.

Keywords: Dialogue Act, Speech Act, Classification, Semantic Similarity, Decision Tree.

1 Introduction

The initial motivation for this work is the development of a technique for Dialogue Act (DA) classification for use in Conversational Agents (CAs). A CA shares the common properties of intelligent agents in exhibiting autonomy, reactivity and pro-activeness [1] and additionally interacts with a human user by means of natural language dialogue [2].

Natural language dialogue involves the exchange of utterances between parties. This work concerns utterances in the form of Short Texts (STs) – which are the length of a typical sentence but do not necessarily obey their grammatical rules. They may arise from entry through a messenger-like text interface or as the output of text-to-speech conversion.

According to DA theory an utterance can be separated out into its propositional content (what the utterance is about) and its DA, which is the function of the utterance in the conversation. There are several taxonomies which decompose DAs into classes. Prior work has reported using between 2 [3] and 232 [4] classes.

The work of John Searle uses 5 classes: Assertive, Directive, Commissive, Expressive and Declarations [5]. This work also adopts a coarse granularity, but one

which is more appropriate for CAs. Specifically, that most dialogue of this type will be composed of questions, instructions and assertions. This assumption is supported by prior work in dialogue systems [6] and robotics [7].

Examples of these DAs are: “Is the door shut?” (question), “The door is shut” (assertion) and “Shut the door!” (instruction). All are derived from the propositional content “door is shut.” Each of the sentences contains the noun *door* and the verb *shut*. The remaining words, *the* and *is* are function words.

Prior work has produced a Slim Function Word Classifier (SFWC) which discriminates effectively between question and non-question DAs. The research question addressed by this study is “Can an SFWC be trained to discriminate between instruction and non-instruction DAs?” This leads to the hypotheses:

H1: An SFWC scores significantly higher than chance in discriminating between Instruction and Non-Instruction DAs.

H0: An SFWC scores no higher than chance in discriminating between Instruction and Non-Instruction DAs.

The level of classification accuracy which would be obtained by chance for the dataset used in this study is 50%. Hypothesis testing uses the corrected re-sampled *t*-test to determine whether the classification accuracy is significantly different from this baseline.

The rest of this paper is concerned with testing these Hypotheses. Section 1 briefly reviews prior work on features for DA classification, function words and choice of classifier. Section 2 describes the experimental procedure including creation of the DA dataset and experiments to train and test classifiers. Section 3 discusses the results and Section 4 the conclusions and recommendations for future work.

1.1 Extraction of Features for DA Classifiers

There are a number of requirements of the features extracted from utterances for DA classification. First, they should actually encode sufficient discriminating information to support accurate classification. Second, they should be capable of being extracted by computationally efficient algorithms. Third, they should be robust to the variations in real-world data.

The most commonly used feature is the *n*-gram, a short sequence of contiguous words. *N*-grams are created by extracting them from a real-world corpus [8, 9]. Any realistically sized corpus will generate large numbers of candidates. Consequently the set of potential *n*-grams is analysed to find a subset which are good predictors for the classification task. For example, [9] used a set of 100 logs from a social interaction computer game containing 4,295 utterances with an average length of 4 words as a training set, using a mutual information measure to find the best 50 predictive unigrams, bigrams and trigrams for 9 distinct DAs.

Cue phrases are another type of feature, similar to *n*-grams, which have been used for DA classification. However, a cue phrase is used in the classifier to predict a single DA class [8, 10]. During training, a cue phrase is selected and assigned the single DA class for which it has the highest predictivity value. During classification if there are two or more cue phrases with the same predictivity the cue phrase with the

highest frequency of occurrence is used, potentially discarding valuable classification information.

Latent Semantic Analysis (LSA) has also been adapted for DA classification [4]. LSA uses terms organised into vectors and the measurement is made between the vectors in a semantic space. LSA is an information retrieval system, so the input text corresponds to an LSA “query” and the DA is that of the closest matching “document”. The LSA semantic spaces are constructed from various corpora, for example *general reading up to 1st year college*. Normally LSA completely discards function words in forming its vectors [11] although this is not clarified in [4].

Some complex combinations of features have been used for DA classification. Keizer et al [3] used length, various starting bigrams, presence of particular words, positive/negative references and presence of particular grammatical classes to classify utterances into two different classes: forward-looking and backward-looking. Forward-looking functions have an effect on following dialogue and backward-looking functions relate to previous dialogue. Verbree et al [12] used a combination of features including the presence of a question mark, a count of the occurrences of the word OR, utterance length, a bigram of the labels of the last two DAs, n-grams of pos-tags for the utterance, n-grams of words in the utterance and the top 10 predictive individual words.

O'Shea et al [13] took a radically different approach to classifying the question DA (vs. non-questions) by using function words only. This method replaced function words by unique tokens and every content word by a common wildcard token.

Performance is difficult to compare, between studies, because there is no established benchmark in this field. Keizer achieved classification accuracy of 76.5% - 88.2% on the SCHISMA corpus. Verbree achieved 59.76% - 89.27%, on the ICSI, Switchboard and AMI corpora. O'Shea achieved an accuracy of 89.43% using a data set collected for the study which was representative of dialogue conducted via the web. Regardless of relative performance, the features used in the work of Keizer and Verbree are complex and computationally intensive to extract, apart from Cue Phrases which may discard considerable useful information.

1.2 Function Words

English words can be partitioned into two classes: Function Words (e.g. articles, prepositions, determiners etc.) and Content Words (nouns, verbs, adjectives and adverbs). An important difference between them is that Function Words are a closed class (the English Language has evolved all of the Function words that it needs and no new words are added to the class). Conversely, new Content words emerge continuously - for example the verb “to tweet” (in the sense of posting a message on Twitter).

There is no definitive list of function words, but they can be extracted from “Stop word” lists which contain mostly function words with a few high frequency content words mixed in. Identifying stop words is important in information retrieval applications where they are removed because they make little or no distinction between documents [11]. One exception is the STASIS measure [14] which recognises the importance of function words because they carry structural information which is useful in interpreting sentence meaning. Again, there is no definitive list of stop words,

although one by van Rijsbergen [15], which is cited frequently contains 250 words. The set of 264 function words compiled for the SFWC [13] is used in this paper. Function words are potentially good features for classification of DAs because they are easily identifiable using small lookup tables, they are stable over time and they are used in all domains so should generalise well.

1.3 Choice of Classifier

A representative range of classifiers has been used in DA classification including Decision Trees [12] Naïve Bayes [3], the Bayesian Network [16] and Artificial Neural Networks [17]. The initial work on the SFWC provided evidence that Decision Trees (DTs) were an effective form of classifier to use with function word features [13] and consequently DTs were chosen for this work.

One of the most well-established DT algorithms is C4.5 [18]. DTs partition the sample space recursively and the outcome is a set of rules induced from a training set of example instances previously labelled with their classes.

The starting point of a DT is one node with all examples labelled as belonging to a single class. In realistic examples this node is “impure” i.e. the examples from the training set contain more than one class. During training, impure nodes are successively partitioned to create new, purer nodes. The final, leaf, nodes are labelled with the appropriate class.

Impure leaves may occur if the attributes have been tested exhaustively without reaching purity. These are labelled with the majority class; consequently test cases from the minority class which reach the leaf during classification result in errors.

There is also the possibility that a tree may model the training data at too high a level of detail, in which case it will not generalise effectively when used with other data. This is prevented by pruning the tree. Confidence Level pruning [18] decides on the level of pruning based on an estimate (derived from the training data) of the likely level of error with other data. Minimum Number of Objects (MNO) pruning [19] removes leaves with less than a specified number of cases and re-distributes them amongst leaves at the level above in the tree.

DTs have two distinct advantages over other classifiers. The first is that the rules “explain” how they reach their decisions very clearly and (combined with pruning) provide a greater insight into the problem domain. The second is that the rules generated by a decision tree can be converted into highly efficient code quite easily for implementation of real-time systems.

2 Experimental Procedure

2.1 Creation of the Dataset

Investigation of the hypothesis that a SFWC can discriminate effectively between Instruction and Non-Instruction DAs requires a suitable tagged (classified) dataset for training and testing the classifier.

There is no standard dataset and those used in studies vary widely using different numbers of DA classes (from 2 [3] to 232 different tags [4]), sizes (81 utterances

[3] to 223,000 utterances [8]) and composition (e.g. one dataset of 2794 DAs was drawn from only two humans [20]). A moderately sized dataset can be used in distinguishing a single DA class from the rest (600 questions and 600 non-questions) [13]. Using 60-fold cross validation with this dataset resulted in over 1000 cases being used for training, in line with a rule-of-thumb suggested by Quinlan [18].

This study uses the questions and non-questions from [13] as the source of non-instructions to be discriminated against. 25 different D.I.Y and self-help sites provided useful sources of instructions, e.g. “Central heating project” and “How to Bisque fire pottery.” The instructions were largely consistent with spoken dialogue.

The initial assumption was that even straightforward instructions would prove difficult to classify because their most salient feature, an imperative verb, is converted to a wildcard. In fact most of the questions and statements begin with a function word so this difference in itself provided information for classification.

This was a similar position to that of the straightforward questions vs. non-questions in [13] where the straightforward questions started with auxiliary verbs or wh-chefts. A similar procedure to that in [13] was adopted to increase the challenge. In this case a set of 1-word preambles was used to push the imperative verb into the second position where the non-instructions were more likely to have a wildcard token and the function words would play a greater role in inferring the occurrence of an instruction. All of the 1-word preambles (e.g. “Again”, “Actually”) were semantically feasible components of instructions, collected from the same sources as the dataset. The outcome was that a set comprised of 50% (600) instructions with 1-word preambles, 25% (300) straightforward questions and 25% (300) straightforward statements was produced to train and test the instruction / non-instruction classifier (examples are shown in table 1).

Table 1. Example training data

Category	Example
FAQ question	Are the Canada pension plan and Canadian old age security benefits taxable?
Blog non-question	She's an insanely talented, sought after calligrapher and one of our dearest long-time friends.
Straightforward Instruction	Do as advised in the handbook where the handbook and our advice appear in conflict.
Instruction with preamble	Again do as advised in the handbook where the handbook and our advice appear in conflict.

For each category, all of the available sentences were preprocessed to restrict the length to a maximum of 25 words, expand contracted forms, tokenise them to extract features for DT construction and remove duplicates occurring due to loss of information in the tokenising process. This was followed by random selection of the number required for each class in the dataset. Decision trees require fixed length records, so an upper limit of 25 words was set (determined empirically based on prior experience of creating real-world CAs).

2.2 Preprocessing the Data

Auxiliary verbs such as *do*, *would* and *can* are function words which play an important role in constructing tenses and expressing conditions such as permission and ability. These words take part in contractions such as *shouldn't*, *can't* and *I'd*. To ensure that these forms were tokenised correctly, the contractions were replaced with their full forms, e.g. *should not*, *can not* before tokenisation. There may be some ambiguity and a brute force solution (e.g. replacing all *I'd* forms with *I would*) has been taken. This approach avoids a potentially computationally intensive parsing stage and did not have a noticeable impact on performance in [13]. All punctuation, apart from apostrophes used in the possessive form, was then stripped out. As the possessive form is a noun variant this is permissible. One consequence is that the presence of a question mark (as used in [12]) is not currently used in this technique.

The preprocessed STs were then tokenised, word-by-word. Each word was looked up in a table of function words which also contained a unique numeric token (range 1-264) for each function word. If there was a match the word was replaced by the relevant token. If the word was not found in the function word table it was assumed to be a content word and replaced by the token 0. If the sentence contained less than 25 words, it was padded out to a length of 25 tokens by inserting “no word present” tokens with a value of 300. Thus the classifier could distinguish very clearly between specific function words, content words in general and empty spaces at the end of sentences in partitioning the problem space. An example of a tokenised instruction is shown in table 2.

Table 2. Tokenisation of training data

Question	Take your multimeter probes and connect them to the outer prongs of the stove's power cord next.
Tokenised form	0,261,0,0,22,0,192,212,190,0,0,148,190,0,0,0,138,300,300,300,300,300,300,300,300,300

The function word table used in this paper contains the words sorted in alphabetical order, so that 1 corresponds to “a” and 264 corresponds to “yourselves.” In the example, the verb “Take” is replaced by 0 because it is a content word. “your” is tokenised as 261, “multimeter probes” is composed of two content words and each is replaced by 0. “And” is replaced by 22, “them” by 192, “to” by 212, “the” by 190, “of” by 148 and “next” by 138. The other content words are replaced by 0s in the same way as “Take”, “multimeter” and “probes.”

2.3 Decision Tree Experimental Trials

This section evaluates the performance of a C4.5 decision tree classifier trained on the dataset (using cross validation) using a range of pruning parameters to identify the best Classification Accuracy and the most compact DTs that can be obtained. Both Confidence Interval (Conf) and Minimum Number of Objects (MNO) were used as pruning techniques. All of the experiments reported in this paper used 60-fold cross validation and were conducted using the Weka data mining package.

The DTs were constructed using the C4.5 version 8 algorithm known as J48 in Weka. Weka also reports significant differences between results in a series against a baseline, using the corrected re-sampled *t*-test with $\alpha = 0.05$. In the following experiments the baseline was the CA achieved by a C4.5 DT using pruning values of Conf = 0.25 and MNO = 2.

2.3.1 Experiments 1 and 2

Experiment 1 examined the effect of pruning at a series of confidence levels running from the default value of 0.25 to 0.05. The first entry, Conf = 0.25 (MNO = 2) sets the baseline for comparison of the other entries. Table 3 shows that increased pruning leads to increasing CA (statistically significant at confidence = 0.15). This reaches a maximum of 89.23 at confidence 0.0775 (39 - 91 nodes).

Table 3. Results of pruning in experiment 1

Conf	0.25	0.2	0.15	0.1	0.05
Classification Accuracy	87.98	88.22	88.87	88.98	88.63
Tree Size	63-121	59-109	51-103	45-97	23-79

Experiment 2 examined the effect of confidence pruning to produce the most compact, efficient tree that did not suffer a statistically significant loss of CA.

Experiment 1 was repeated, using pruning confidence intervals running from 1.0 E-3 to 1.0 E-5. Table 4 shows the results.

Table 4. Results of pruning in experiment 2

Conf	1.0E-3	5.0E-4	1.0E-4	5.0E-5	1.0E-5
Classification Accuracy	87.92	87.92	87.90	87.89	87.87
Tree Size	19-27	19-27	19-23	19-23	15-23

Execution was interrupted at the 5.0E-6 confidence level, presumably due to a floating-point arithmetic error. This was before a significant drop in CA below the baseline was reached.

2.3.2 Experiments 3 and 4

Experiment 3 examined the effect of pruning at a series of MNO levels running from the default value of 2 to 20 (table 5). This pruning did not discover a significant change in CA. The maximum CA obtained with MNO pruning was 88.03% at MNO = 4, which was not significantly better than the baseline value.

Table 5. Results of pruning in experiment 3

MNO	2	5	10	15	20
Classification Accuracy	87.98	87.68	86.98	87.03	86.91
Tree Size	63-121	39-91	27-53	19-33	15-29

Experiment 4 examined the effect of MNO pruning to produce the most compact, efficient tree that did not suffer a statistically significant loss of CA.

Table 6. Results of pruning in experiment 4

MNO	36	37	38	39	40
Classification Accuracy	87.29	87.19	86.98	86.43	85.49
Tree Size	11-21	11-21	11-21	7-21	7-19

Table 6 shows the results of pruning using MNO levels running from 36 to 40 instances. At MNO = 40 the reduction in CA from the baseline was significant. Pruning at MNO = 39 gave a good combination of tree size and CA.

3 Discussion of Results

Confidence level pruning achieved a CA of 89.23% at confidence = 0.0775 and MNO pruning achieved a CA of 88.03% at MNO = 4. The average baseline tree size was 93 nodes. It was not possible to discover the optimum pruning level with confidence-based pruning, but pruning at MNO = 39 produced an average tree size of 15. The maximum depth of nesting of this tree was 5, suggesting that efficient implementation would have a low computational overhead.

4 Conclusions and Future Work

The results of the experiments provide good evidence to support the alternative hypothesis that an SFWC scores significantly higher than chance in discriminating between Instruction and Non-Instruction DAs. Statistically significant results (using the corrected re-sampled *t*-test) for CA well above baseline have been obtained. The reduction in tree size from pruning provides further strong evidence that true generalisation is taking place.

The pruned decision trees combine function word features ranging from the 1st to the 6th word. This indicates that the technique is exploiting more than simple n-gram information and that the function words in the vicinity of the imperative verb are key features in classification.

When these results are considered with those from [13], good evidence is accumulating to support the general principle of using function words for DA classification.

Future work will involve the development of specialist classifiers for particular variants of the instruction form, which could be combined into a single multiclassifier. It will also investigate the possibility of producing a gold standard benchmark dataset from the DA training and testing datasets produced. Additionally, experiments will be conducted on optimising the encoding of features. This is expected to reduce top level splitting and preserve information for better splitting at later stages in the tree. It is also expected to improve generalisation by allowing a particular route through the tree to represent more cases.

References

1. Wooldridge, M., Jennings, N.R.: Intelligent Agents: theory and practice. *The Knowledge Engineering Review* 10(2), 115–152 (1995)
2. Bickmore, T. and Cassell, J.: 'How about this weather?' Social Dialog with Embodied Conversational Agents. In: *The American Association for Artificial Intelligence (AAAI) Fall Symposium on "Narrative Intelligence"*, Cape Cod, MA (2000)
3. Keizer, S., op den Akker, R., Nijholt, A.: Dialogue Act Recognition with Bayesian Networks for Dutch Dialogues. In: *Third SIGdial Workshop on Discourse and Dialogue*, Philadelphia (2002)
4. Serafin, R., Di Eugenio, B., Glass, M.: Latent Semantic Analysis for dialogue act classification. In: *The 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada (2003)
5. Searle, J.R.: *Mind, Language and Society*. Weidenfield & Nicholson (1999)
6. Crockett, K., et al.: Bullying and Debt: Developing Novel Applications of Dialogue Systems. In: *Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)*, Pasadena, CA (2009)
7. Längle, T., et al.: KANTRA — A Natural Language Interface for Intelligent Robots. In: *Intelligent Autonomous Systems (IAS 4)*, Amsterdam (1995)
8. Webb, N., Hepple, M., Wilks, Y.: Dialogue Act Classification Based on Intra-Utterance Features. In: *AAAI 2005*, AAAI Press, Pittsburgh (2005)
9. Orkin, J., Roy, D.: Semi-Automated Dialogue Act Classification for Situated Social Agents in Games. In: *The AAMAS Agents for Games & Simulations Workshop* (2010)
10. Webb, N., Liu, T.: Investigating the Portability of Corpus-Derived Cue Phrases for Dialogue Act Classification. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester (2008)
11. Deerwester, S., et al.: Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science* 41(6), 391–407 (1990)
12. Verbree, D., Rienks, R., Heylen, D.: Dialogue-Act Tagging Using Smart Feature Selection; Results On Multiple Corpora. In: *IEEE Spoken Language Technology Workshop* (2006)
13. O'Shea, J., Bandar, Z., Crockett, K.: A Machine Learning Approach to Speech Act Classification Using Function Words. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6071, pp. 82–91. Springer, Heidelberg (2010)
14. Li, Y., et al.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering* 18(8), 1138–1150 (2006)
15. van Rijsbergen, C.J.: *Information Retrieval*. Butterworths, Boston (1980)
16. Stolcke, A., et al.: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics* 26, 339–373 (2000)
17. Kim, K., Kim, H., Seo, J.: A neural network model with feature selection for Korean speech act classification. *Int. J. Neural Syst.* 14(6), 407–414 (2004)
18. Quinlan, J.R.: *C4.5: programs for machine learning*. Morgan Kaufmann series in machine learning. Morgan Kaufmann Publishers, San Mateo (1993)
19. Witten, I.H., Eibe, F.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Elsevier, San Francisco (2005)
20. Venkataraman, A., Stolcke, A., Shriberg, E.: Automatic Dialog Act Labeling With Minimal Supervision. In: *9th Australian International Conference on Speech Science and Technology* (2002)

An Application of Pattern Matching Stemmer in Arabic Dialogue System

Mohammad Hijjawi, Zuhair Bandar, Keeley Crockett, and David Mclean

School of Computing, Mathematics and Digital Technology,
Manchester Metropolitan University, Chester Street M1 5GD
Manchester, United Kingdom
{M.hijjawi, Z.bandar, K.crockett, D.mclean}@mmu.ac.uk

Abstract. This paper proposes the building of a stemmer for the Arabic language. This stemmer is largely based on pattern matching and pattern strength techniques. Stemmers are algorithms to extract root from a word by removing its affixes. Stemming has been applied for large number of applications, such as: indexing, information retrieval systems, and web search engines. This paper will also proposes the application of stemming as a pre-processing stage in a dialogue system (DS). The proposed stemmer was compared with three other well known stemmers and achieved favourable accuracy.

Keywords: Arabic, Stemming, Morphological analyser, Pattern Matching, and Dialogue system.

1 Introduction

In Arabic language, a root is the initial form of a word that cannot be analysed further. Affixes such as prefixes, infixes, and suffixes can be added before, inside, or after a root respectively to generate more meaningful words. This addition might replace letters or even remove letters from the root especially in roots which contain vowel letters [1]. Stemming is a computational process which removes all kind of affixes from a word to extract its root [2]. This process will reduce the size of word variations to its minimum. And thus, decreasing the searching time and improving the performance of a number of applications such as information retrieval systems, text-based searching applications [3] and dialogue systems as this paper proposes. A dialogue system or conversational agent is a computer program to simulates intelligent dialogues with human [4].

Conversational agent might be applied in many applications such as: education, information retrieval, business, help desk, entertainment and e-commerce [5]. As discussed earlier that stemming is of great benefit when used with text-based searching applications. This is especially useful when used with pattern matching based Conversational Agents. Scripting such agents involve the generation of a very large number of patterns. Utilising stemming in such application, as a pre-processing stage, will reduce the search time greatly. However, the proposed application of stemming in this paper is novel and to our knowledge no other DS (Conversational Agent) has

employed stemming for scripting or as a pre-processing stage. In addition, a little or no research has been done on conversational agents based on Arabic. Therefore, this paper is a subsequent work on developed conversational agent based on Arabic language.

Over the years, a number of stemming approaches have been built for different languages including Arabic. However, an approach which might be useful for one language may not be useful for another due to the features differences among languages. Generally, these approaches are classified into: manual dictionary construction, light stemmers, morphological analysis, and statistical stemmers [2].

Manual dictionary construction is an early approach which works on small text collections to build dictionaries of roots and stems manually. This approach is impractical due to its large sized corpora [6]. However, Arabic Light stemmers perform a light stripping off process by removing prefixes and suffixes to generate a stem [1]. Nevertheless, Arabic morphological approaches deal with all types of affixes. Therefore, they are attempting to remove prefixes and suffixes to generate the stem and continue with infixes to generate the root [1]. Finally statistical stemmers, attempt to group word variations using clustering techniques to extract the root without dealing with any linguistic issues [2].

Arabic stemming still faces many challenges [7]. It is one of the semitic languages which have rich inflectional and derivational features. In Arabic, affixes can reform a word to reflect a tense, a gender (masculine, feminine) and/or number (singular, dual, plural) which increase the stemming challenge as shown in the examples in table 1. Moreover, there are no standard lists for these affixes due to the high derivational and inflectional features in Arabic.

Table 1. Represents examples of some Arabic roots transformations

Arabic Word	English meaning	Reflect Tense	Reflect Gender	Reflect Number	Root
كتاب	Book	Noun	N/A	singular	كتب
كتابين, كتابان	Two books	Noun	N/A	dual	كتب
كتب	Books	Noun	N/A	plural	كتب
كتب	He wrote	Past	Male	singular	كتب
كتابك	Your book	Noun	N/A	singular	كتب
كتبكم	Your books	Noun	Male/Female	plural	كتب
كتبهم	Their books	Noun	Male	plural	كتب
كتبهن	Their books	Noun	Female	plural	كتب
يكتب	He is writing	Present	Male	singular	كتب
تكتب	She is writing	Present	Female	singular	كتب
كتبت	She wrote	Past	Female	singular	كتب
ستكتب	She will write	Future	Female	Singular	كتب

In addition, the roots that contain one or more vowel letters have no fixed rules in word's transformation process. In other words, vowel letters might be replaced with other letters or even removed during the transformation process. Furthermore, the stop words which represent a 40% of Arabic text [8] can accept affixes which might

generate a new words with different meanings. And thus, there is no standard list for stop words in Arabic.

Another challenge is dealing with nouns and foreign arabised words that might come from different languages. For instance, unlike English, does not support capitalisation. Consequently, it is difficult to differentiate proper nouns [9]. Most of text-based searching systems which use stemming ignore the stop words due to their ineffectual results. Because there are no roots for these words, they are usually removed from the targeted stemmed words.

Presently, different types of Arabic language could be used in the daily conversations between people. Classical Arabic, which is used in the holy Quran, is more complex in its grammar and vocabularies than modern Arabic. Modern Arabic is formally recognised as the official language for Arab countries. Modern Arabic is used in every day language, in the media, education, and literature. Instead, they are use a third type which is known as Colloquial Arabic. Colloquial Arabic is less sophisticated than modern type and due to its simplicity, most people use it in their every day conversations [10].

Arabic grammar has many powerful structures, for instance words patterns. Words patterns are words formed based on affixation and vocalisation processes on the Arabic roots. They can be classified into noun patterns which represent nouns and verb patterns which represent verbs [1]. For instance the word (عامل) belongs to pattern (فاعل), and (يعمل) belongs to pattern (يفعل). Most of the existing Arabic root-based stemmers use words patterns in order to extract the root [11].

A number of well known Arabic stemmers use a root dictionary during the stemming process such as Khoja stemmer in order to validate the resulted root [11]. However, using the pre-built root dictionary in stemming with online applications will increase the overall processing time. Moreover, root dictionaries are manually constructed and therefore they are hard to maintain.

The rest of this paper is organised as following: section 2 present previous works; section 3 explains the proposed methodology. The evaluation is reported in section 4. Using the stemming in DS is illustrated in section 5. Finally, the conclusion of this paper is presented in section 6.

2 Previous Research in Arabic Stemmers

A number of different approaches have been used to implement stemming for the Arabic language. Regardless of the used approach, these stemmers share a common problems and tend to generate errors in several forms; which are over-stemming, mis-stemming and under-stemming [3]. Over-stemming occurs when two words with different roots are stemmed back to the same root. Under-stemming occurs when two words that should be stemmed to the same root and are not. Mis-stemming occurs when removing some of the original root's letters by assuming they are affixes. In addition, trying to overcome one error might increase other error types [3]. Nevertheless, these errors might have occurred due to the blind way of stripping off affixes. In the following sections, three well known Arabic stemmers Khoja stemmer [11]. Buckwalter Morphological analyzer [12], and Tri-literal Root Extraction stemmer [13], will be briefly discussed. The proposed stemmer will be compared to these stemmers at the evaluation stage.

2.1 Khoja Stemmers

Khoja's stemmer starts with a pre-processing stage to remove diacritics, numbers, punctuation, and definite articles. In addition, the processed word is then checked to establish whether or not it is a stop word. Stop words are removed from the text. Longest suffix and longest prefix are then stripped off in order to match the remaining word with word patterns to extract the root. Finally, the extracted root will be checked with a list of valid roots (dictionary). If a match is found, it will return the root, otherwise returns the original word [11]. Khoja's stemmer relies on the first matching occurs. As a result, no differentiation is made between matches which might result in the selection of an inappropriate match.

2.2 Tim Buckwalter Morphological Analyzer

The analyser uses lexicons of Arabic prefixes, stems, and suffixes. It uses compatibility tables to validate the pairs: prefix-stem, stem-suffix, and prefix-suffix. These pairs will help the analyser to start segmenting the transliterated word and return the possible segmentations of this word using English characters [12].

2.3 Tri-literal Root Extraction Algorithm

This stemmer relies on assigning weights for word's letters and then multiply these weights by the letter position. All affixes in Arabic can be represented using the combinations of the word (سألتمونيها) letters. Different weights assigned to the letters of this word to differentiate affixes letters and assigning a weight of zero to consonants letters. It then retrieves the letters of the lowest weights and considers them as root letters [13]. However, one or more letters of the mentioned word (سألتمونيها) might be originally in the root which might leads to the generation of unrelated roots.

3 The Proposed Arabic Stemmer - Methodology

The proposed Arabic stemmer uses pattern matching to strip off the affixes. The matching process matches a target word against a pattern and then extracts the matched letters as a result. This pattern might be a prefix pattern, suffix pattern, or Arabic word pattern. If there is matching, the matcher returns the extracted letters from the matching process; otherwise, it will return an empty string. However, the matching process will also rely on the matching strength so matches are distinguished with each others and the best match is then selected. Selection of the best match here means the selection of the most accurate representation of the word. When a word matches a prefixes or suffixes, the selection of the lowest pattern strength will be selected. But when a word matches a word patterns, the highest strength will be selected. The matching selection relies on the following pattern matching strength.

$$\text{Pattern Strength} = (2 * \text{ClearEnds}) + \text{Pattern Fixed Length}. \quad (1)$$

Where,

- * ClearEnds = 0, if the pattern does not start or end with Arabic letter.
- Or ClearEnds = 1, if the pattern either start or end with Arabic letter.
- Or ClearEnds = 2, if the pattern start and end with Arabic letter.
- * Pattern Fixed Length = Number of Arabic letters in the pattern.

The following example illustrates the effect of the pattern matching strength on the matcher selection: The word (اعمال) matches the pattern (%%%) which transformed from the Arabic word pattern (افعال) and using the above equation the pattern strength is 4. This word will also match the pattern (%%%) which can be transformed from (افعل) which will get the strength of 3. Thus, the first pattern will be chosen due to its higher strength. Using these techniques will enable the stemmer to overcome a major weakness associated with other stemmers by differentiating between different matches and select the best match.

Proposed Arabic stemmer needs a pre-processing stage before starting the stemming process, which will be discussed in section 3.1.

3.1 The Proposed Arabic Stemmer - Pre-processing

1. Generate a Prefix List: convert the prefix list to pattern prefix list in order to match the pattern prefixes with the word. Then extract the matched letters. For instance: convert the (ال) to (*ال) and (و) to (*و), where (*) is a wildcard which matches any Arabic letter.
2. Generate a Suffix List: convert the suffix list to pattern suffix list for the same reason as above. For instance: convert the (ك) to (ك*) and (ين) to (ين*), where (*) is a wildcard which matches any Arabic letter.
3. Create a Stop Word or functional words list.
4. Generate an Arabic Word Patterns List: convert the Arabic word patterns list to a matchable patterns list in order to extract the root letters. For instance: convert the word pattern (مفعل) to (%%%) and (مفعله) to (ه%%%) where (%) is a wildcard which matches a single Arabic letter.

3.2 Arabic Stemmer Algorithm

The stripping off process ultimately depends on the matching process. The result of matching process is the extracted letters or empty string if there is no match. Moreover, the stripping off process works recursively until the root is reached. The following steps explain the proposed stemmer:

Step 1: Tokenising the targeted text into words.

Step 2: Pre-processing stage in order to remove the punctuations, numbers, and Arabic diacritics from the tokenised word.

Step 3: Normalisation process to convert confused letters or different forms for the same letter to one letter. For instance: converting letters (أ, إ, ؤ) into (أ) letter and converting letters (ي, ي, ي) into (ي) letter.

Step 4: Check the processed word's length, if it is >3, continue to the next step; otherwise return the processed word as a root.

Step 5: Check the processed word with the stop words list; if it is not a stop word continue to the next step; otherwise keep it as it is.

Step 6: check if the processed word contains prefix or suffix, by matching it with the word patterns using the following functions:

Result1 = Matching (processed word, word-pattern list).

Result2 = Matching (processed word, prefix list).

Result3 = Matching (processed word, suffix list).

Where, matching function matches the processed word with a pattern and the result is the extracted matched letters or an empty string if there is no match.

Step 7: If Result1 is not empty, continue to the next step; otherwise, jump to step 9.

Step 8: Keep matching the extracted word with word patterns until the root of length 3 is reached or return the best last extracted root.

Step 9: Apply Result2 and Result3 to the following rules:

If Result2 is not empty and Result3 is empty; jump to step 10.

If Result2 is empty and Result3 is not empty; jump to step 11.

If Result2 is not empty and Result3 is not empty; jump to step 12.

If Result2 is empty and Result3 is empty; jump to step 13.

Step 10: Keep matching the extracted word with the prefix list and then match the best result with the word pattern list. Repeat the process until the root of length 3 is reached or return the result of last best match.

Step 11: Keep matching the extracted word with the suffix list and match the best result with the word pattern list. Repeat the process until the root of length 3 is reached or return the result of last best match.

Step 12: This rule deals with prefix and suffix in the processed word.

Step 12.1: keep matching the extracted word with suffix list and match the best result with word pattern list and repeat the process until the root of length 3 is reached. If not, jump to step 12.2.

Step 12.2: check if the length of the previous result (step 12.1) is greater than 4; keep matching the extracted word with prefix list and then match the best result with the word pattern list and repeat the process until the root of length 3 is reached or return the result of last best match.

Step 13: This rule deals with word which does not have prefix or suffix and does not match any word pattern. Thus, this word might be a foreign Arabised word or noun. Therefore, the stemmer will keep it as it is by assuming that there is no root for this word.

4 Experiments

4.1 Experiments Methodology

The proposed stemmer was compared with other three well-known Arabic stemmers, Khoja [11], Buckwalter [12], and Tri-literal Root extraction algorithm [13]. These Arabic stemmers were evaluated in [7] for two types of Arabic, classical and modern.

The authors of the selected aforementioned study [7], kindly provided us with the tested modern Arabic dataset with its gold standard document which our experiments will rely on this type of Arabic only because our DS based on this type.

4.2 Experimental Results

By running our stemmer on the selected dataset (1005 words), two accuracy measurements has been selected and applied in two experiments. First experiment has been done by including the stop words and the second one by excluding them. The correctness accuracy results were evaluated and compared with the roots in the gold standard document. Tables 2 and 3 show the reported accuracies for the first and second experiments respectively.

4.3 Experiment 1

Table 2. Accuracy - including stop words

Stemming Algorithm	Accuracy
Khoja Stemmer	77.01 %
Tri-Literal Root algorithm	76.72 %
Tim Buckwalter morph. Analyzer	40.70 %
The proposed Stemmer	78.6%

4.4 Experiment 2

Table 3. Accuracy - excluding stop words

Stemming Algorithm	Accuracy
Khoja Stemmer	72.3 %
Tri-Literal Root algorithm	64.37 %
Tim Buckwalter morph. Analyzer	39.3 %
The proposed Stemmer	73.34 %

4.5 Discussion

Results in tables 2 and 3, show that our proposed stemmer has outperformed the other three stemmers for both data sets. Khoja's stemmer gave the closest results to our stemmer. This is likely because the proposed stemmer shares many features with Khoja. However, the recursive way of stripping off the affixes and the application of matching strength to select the best match has paid off.

5 The Application of Stemming in Dialog Systems

Dialogue systems (DS), especially those based on pattern matching, suffer from the large number of patterns that are required to be scripted to build a coherent domain. In high rich inflectional and derivational language such as Arabic this problem will be further complicated due to the existence of infixes and the replacement process of some letters during the root transformation.

The scripter should take into consideration every expected change that could occur on the pattern's keywords when he/she script them. Using the wildcards to match the affixes is the current solution for patterns building, the scripter should mind the existence of prefix alone, infix alone, and suffix alone, which might generate at least 4 patterns by assuming the pattern has just one keyword. However, when the pattern has two keywords, at least 16 patterns might generated to handle the expected conversation. Moreover, matching all affixes types in one pattern, to reduce the number of scripted patterns, will lead to the production of a very weak pattern.

Another challenge, which the scripter will have to deal with, is that of infixes which might change the word's form by replacing or removing some letters in case their roots contain one or more vowel letters. In addition, using wildcards to match affixes might lead to the loss of needed fragment of text.

The application of stemming as a pre-processing stage in DS by converting all word variations to word's root will reduce the amounts of scripted patterns to the minimum. Instead of scripting all patterns keyword forms, the scripter will only need to script the root. When the DS receives a user's utterance, it starts the stemming process to convert the utterance's words to its roots forms as illustrated in the proposed stemming methodology in section 3.2.

Furthermore our stemmer will keep all the stop words which might be need by the dialogue system to reformulate the original utterances.

6 Conclusion

This paper presented a novel technique of stemming for the Arabic language. Results on two data sets showed that the proposed stemmer has outperformed other well established stemmers. Our stemmer proved that changing the traditional way of stripping off the affixes and that using the pattern strength to decide the best match instead of relying on the first match has improved the accuracy.

Acknowledgment

The authors would like to express their appreciation to Mr. Majdi Sawalha (University of Leeds, UK) for providing the experiment's materials.

References

1. Hayder, K., Al Ameen, S.O.A.K., Amna, A., Kaabi, A., Khadija, S., Shebli, A., F. N.H.A.N.N., Shamsi, A., Shaikha, S., Muhairi, A.: Arabic Light Stemmer: A New Enhanced Approach
2. El-Khoribi, R., Ismael, M.: An intelligent system based on statistical learning for searching in arabic text. ICGST International Journal on Artificial Intelligence and Machine Learning (2006)
3. Eiman Tamah, A.-S., Jessica, L.: Towards an error-free Arabic stemming. In: Proceeding of the 2nd ACM Workshop on Improving Non English Web Searching. ACM, New York (2008)

4. O'Shea, K., Bandar, Z., Crockett, K.: A Novel Approach for Constructing Conversational Agents using Sentence Similarity Measures (2008)
5. Abu Shawar, B., Atwell, E.: Chatbots: Are they Really Useful? (2005)
6. Al-Kharashi, A.I., Evens, M.: Comparing words, stems, and roots as index terms in an Arabic Information Retrieval System. *J. Am. Soc. Inf. Sci.* 45(8), 548–560 (1994)
7. Sawalha, M., Atwell, E.: Comparative evaluation of arabic language morphological analysers and stemmers. In: *Proceedings of COLING 2008 22nd International Conference on Computational Linguistics* (2008)
8. Sawalha, M., Atwell, E.: *توظيف قواعد النحو والصرف في بناء محلل صرفي للغة العربية*. An application of grammar in building morphological analyzer for Arabic
9. Diab, M., Hacioglu, K., Jurafsky, D.: Automatic tagging of Arabic text: from raw text to base phrase chunks. In: *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, Boston (2004)
10. Al-Saidat, E., Al-Momani, I.: Future markers in modern standard arabic and jordanian arabic: A contrastive study. *European Journal of Social Sciences* 12 (2010)
11. Khoja, S.: *Stemming Arabic Text* (1999), <http://zeus.cs.pacificu.edu/shereen/research.htm>
12. Buckwalter, T.: official web site, <http://www.qamus.org>
13. Al-Shalabi, R., Kanaan, G., Al-Serhan, H.: New approach for extracting Arabic roots. In: *International Arab Conference on Information Technology (ACIT 2003)*, Egypt (2003)

Information Extraction for SQL Query Generation in the Conversation-Based Interfaces to Relational Databases (C-BIRD)

Majdi Owda, Zuhair Bandar, and Keeley Crockett

The Intelligent Systems Group, School of Computing, Mathematics and Digital Technology,
The Manchester Metropolitan University, Chester Street,
Manchester, M1 5GD, UK
{m.owda, z.bandar, k.crockett}@mmu.ac.uk

Abstract. This paper presents a novel methodology of incorporating Information Extraction (IE) techniques into an Enhanced Conversation-Based Interface to Relational Databases (C-BIRD) in order to generate dynamic SQL queries. Conversational Agents can converse with the user in natural language about a specific problem domain. In C-BIRD, such agents allow a user to converse with a relational database in order to retrieve answers to queries without knowledge of SQL. A Knowledge Tree is used to direct the Conversational Agent towards the goal i.e. creating an SQL query to fit the user's natural language enquiry. The use of IE techniques such as template filling helps in answering the user's queries by processing the user's dialogue and extracts understandable patterns that fills the SQL templates. The developed prototype system increases the number of answered natural language queries in comparison to hardcoded decision paths in the knowledge trees.

Keywords: Information Extraction, Knowledge Trees, Conversational Agents, and Conversation-Based Interfaces to Relational Databases.

1 Introduction

The development of reliable Natural Language Interfaces to Databases (NLIDBs) can accelerate the progress of interactive applications. A considerable amount of literature has been published on this area. Androutsopoulos [21] defined four main approaches to NLIDBs, each approach is supported by a specific architecture.

The first approach is based on Pattern-Matching which is one of the first architectural techniques that have been used for the design and development of NLIDBs. An example of such a system is SAVVY [22]. In order to make NLIDBs more reliable this approach utilizes a highly structured semantic grammar, rapid authoring tools, and a query paraphraser. However, in large databases a great number of patterns have to be created. Furthermore, the process of identifying relevant patterns is critical and time consuming. The second approach is based on an Intermediate Language, which is used to represent an intermediate logical query generated from a natural language question. The intermediate logical query is then transformed into SQL. Two such

systems of this category are MASQUE/SQL [5] and EDETE [3]. The main disadvantage of this approach is that failure to generate an SQL query will result in a response which is not informative to the user.

The third approach is based on a Syntax-Based Family of Architectures, where the natural language question is syntactically analyzed to create parse tree(s). The resulting parse tree(s) will then be used directly to create database queries. An example of such a system is LUNAR [8]. However, a major difficulty of this approach is in generating the mapping rules from the parse tree to SQL. The fourth approach is based on the Semantic-Grammar Family of Architectures. This approach also involves the construction of a parse tree and mapping of the parse tree to SQL. The difference between this approach and syntax based approach is the use of semantic categories instead of syntactic concepts where non-leaf nodes in the parse tree are semantic categories. The semantic information about the knowledge domain is hard-wired into the semantic grammar. This family includes systems such as: PLANES [7, 21] and LADDER [21]. This approach appears to work well in restricted domains (e.g. small databases). However, creating new system will involve creating a new semantic grammar which is both time consuming and complex.

Currently there is some ongoing work on NLIDBs which does not fall into the categories defined by Androutsopoulos such as Precise [1], Step [6] and C-Phrase [19].

A reliable NLIDB system should overcome the problems associated with the above approaches. Firstly any such system should provide adequate coverage of patterns to represent the domain. Secondly, the system should allow for interactivity with the user. The user must be able to understand the system responses in the case of failure of generating SQL query. Thirdly, overcoming the problem of generating mapping rules from syntax based parse tree to SQL.

The Conversation-Based Interfaces to Relational Databases [1] is based on the pattern-matching architecture but has additional components: knowledge trees, a goal oriented conversational agent and information extraction module. Knowledge trees are used to structure the domain knowledge to cover common queries in a particular domain. Whereas a goal oriented conversational agent provides the natural language interface and can help to disambiguate the user's queries, by allowing for dialogue interaction. The use of the knowledge trees helps direct the conversational agent towards the goal (i.e. query generation from natural language) through interaction with the user. The information extraction module helps extract understandable patterns from the relational database and project the created templates into SQL queries. The proposed approach reduces the complexity of building NLIDBs.

This paper is organized as follows: Sections 2, 3 and 4 will introduce the main concepts behind Conversational Agents, Knowledge Trees, and Information Extraction. Section 5 will introduce the Enhanced Conversation-Based Natural Language Interfaces to Relational Databases Framework. Section 6 presents the results. Section 7 includes the summary.

2 Conversational Agents

A conversational agent is human-computer dialogue system that interacts with the user turn by turn using natural language. Conversational agents have been used in

many areas (e.g. help and advice, tourism, information retrieval, and learning styles [15, 16, 17, 18]). This field started by Alan Turing in 1950 with his famous paper Computing Machinery and Intelligence [2]. In the 1960s work on conversational agents included ELIZA [16] which is a computer program simulates a psychotherapist. ELIZA is based on simple stimulus-response architecture (i.e. patterns and their responses), where ELIZA turns the input patterns into questions as responses. A number of programs have been developed based on lessons learned from ELIZA such as: PARRY [9], ALICE [14], and ADAM [15]. PARRY was built for a specific task that simulates paranoid patients, which makes PARRY hard to be adopted as autonomous component. ALICE and ADAM are general purpose conversational agents. They can be trained and adopted as autonomous components. Allen [17] characterized conversational agents, based on the implementation technique used, into five different groups (ranging from least to most complex): Finite-state script, Frame-based, Set of Contexts, Plan-based models, and Agent-based models.

The conversational agent in our framework should have a set of requirements in order to develop a Conversation-Based NLIDB. Firstly it should be an autonomous general purpose conversational agent. Secondly it is capable to deal with a set of contexts, and to direct the users towards a goal, which gives the conversational agent the flexibility and robustness. Thirdly it should be able to converse freely within a specific domain. Finally it should be able to extract, manipulate, and store information from the users.

ALICE and ADAM meet the above requirements. But ADAM is also a goal oriented which is more suitable to develop a reliable Conversation-Based NLIDB. The proposed system will be using ADAM and will have the advantage of having a user friendly interface that speaks the user's language and allows for dialogue interaction.

3 Knowledge Trees

Knowledge Trees has been used for Knowledge Representation and maintenance in many systems (e.g. ESTA [10], AMRAPALIKA [6] HWYCON [20]). The main purpose of using such trees for Knowledge Representation is that its structure is similar to the way in which humans represent knowledge. The domain expert is not required to have experience in any programming language. The tree structure also provides an easy way to revise and maintain the knowledge base.

A variety of definitions of the term knowledge tree have been suggested. Yager [12] defines the idea of a knowledge tree as a mechanism providing a global framework for the mobilization of a knowledge base in response to a specific query. Our definition of a knowledge tree in the context of this paper is a tree where the knowledge is organized in a hierarchical structure based on the expert knowledge which has been extracted and developed by a knowledge engineer. Hence the knowledge trees serves as a road map for the conversation flow in a specific domain. For example, consider the knowledge tree in Figure 1. Assume T_0 is the start node, and T_1, T_2, T_3 are sub-nodes for the start node, and T_4, T_5 are sub-nodes for node T_2 .

T_0, T_2, T_3 are direction nodes which direct the conversation towards the formulation of an SQL query. T_1, T_4, T_5, T_6, T_7 are report nodes which provide the syntax of the SQL queries. The direction nodes include patterns to carry on a conversation and

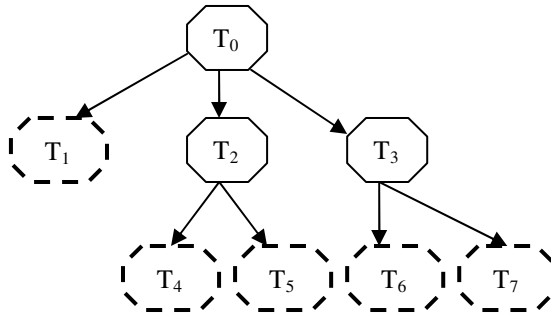


Fig. 1. Simple knowledge Tree

directions to sub-nodes. The report nodes where they carry the SQL queries and they might include patterns in case further information needed.

4 Information Extraction Techniques

Information Extraction (IE) is a cascade of sequence of steps at each step the system will add a structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically. There are two basic approaches to the design of IE systems. Firstly, Knowledge Engineering Approach which is based on having a knowledge engineer that keep maintaining the system rules. Secondly, Automatic Training Approach where is no need for someone to write rules for the IE system, just someone who knows well about the domain and then the task is to take a corpus of texts and annotate the text appropriately for the information being extracted.

According to Appelt and Israel [13] rule based approach is best when the resources such as lexicons, and rule writers are available. However training data is expensive to obtain, extraction specification is likely to change, and the highest possible performance is critical. The rule based approach will help with the problem of generating SQL queries from natural language. The relational databases have rich semantic structure since there have been an engineering effort invested already during the development period. Using Rule Based Information Extraction to project the Natural Language Query into an SQL Statement has been made possible.

5 Enhanced Conversation-Based Natural Language Interfaces to Relational Databases Framework

Figure 2 shows the proposed framework architecture, which has been divided into three main levels; each level has a group of modules associated with it.

The following sections explain the supportive and backend levels first and then the interactive level, since the interactive level depends on both the supportive level and backend level.

5.1 The Supportive Level

This intermediate level contains modules that support the framework functionality, it contains the following components: The Knowledge Tree, Conversational Agent, Semantically Understandable Relational Database Patterns, SQL Queries Templates and Tree for dynamically generated queries.

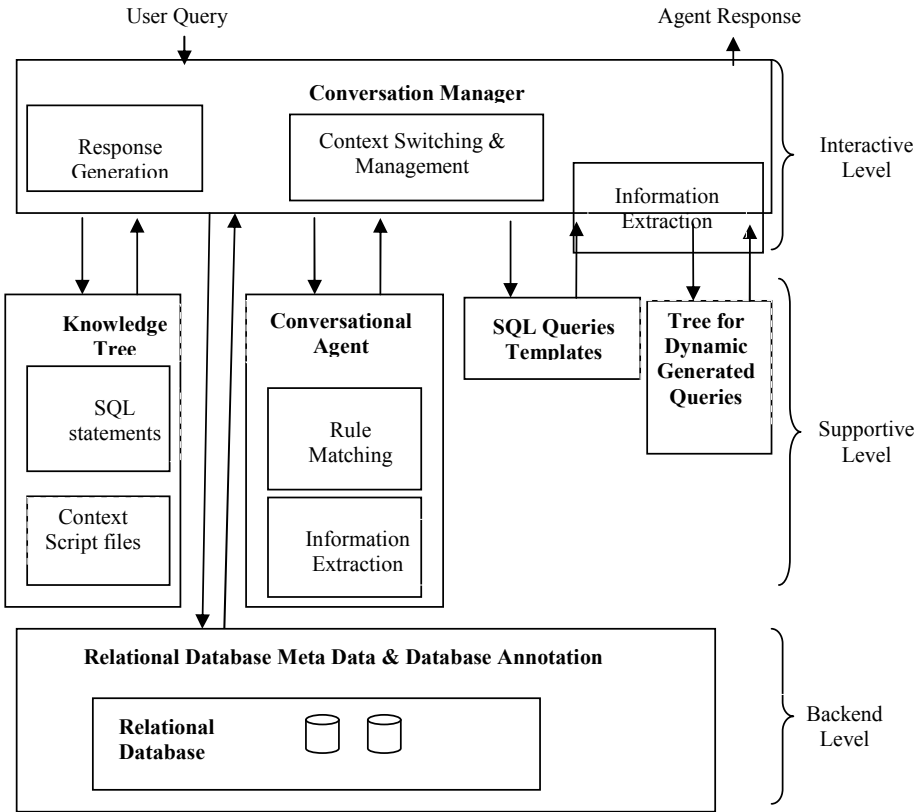


Fig. 2. Shows Conversation Based Interface to Relational Databases (C-BIRD)

5.1.1 Knowledge Tree

The role of the Knowledge Tree is mainly for structuring knowledge elicited by knowledge engineers based on the expert knowledge. In this enhanced framework the knowledge tree is not going to be the only source for creating an SQL query, due to the fact that there will be an algorithmic solution which will support the framework when the knowledge tree can't support the conversation. The knowledge tree will help answer common queries, setting the grounds and initial information extraction through the tree nodes that will help later with trying to formulate a dynamic SQL query.

5.1.2 Conversational Agent

The Conversational Agent in this enhanced framework has been equipped firstly, with more scripts to extract and engage the user with more conversations. Secondly, the Conversational Agent will get the benefit of using the Information Extraction Module that utilizes the Semantically Understandable Relational Database Patterns to fill the SQL Templates in order to generate automated responses i.e. responses based on untrained utterances.

5.1.3 Semantically Understandable Relational Database Patterns

The relational databases have rich semantic structure because there has been an engineering effort invested already during the development period.

The idea behind introducing semantically understandable relational database patterns in order to build Conversation-Based Interface to Relational Databases (C-BIRD) is of great value, which can be utilized by the Conversational Agent. The use of Conversational Agents enable us to converse with the user about a certain context. In the database area the contexts change once you move from table/database to another. Defining the main database elements to the Conversational Agent will help in laying the ground for the Conversational Agent to carry out a conversation within/around an understandable pattern. For example, the table name called “**customer_dim**” will be understood as an understandable pattern “**customers**” table.

The following database elements have been identified as the major understandable patterns in the relational database to be annotated and used in the information extraction module (Table Names, Column Names, Column Values, and Functions Names). The Information Extraction module is going to extract these understandable patterns and use them to fill the SQL queries templates and then generate responses accordingly.

The current prototype has been developed for one database only, which clearly did not include the database name as a relational database understandable pattern. In the case of having more than database, the database name will be included. The module in section 5.2.1 named Relational Database Annotation & Meta Data Module will introduce the annotations of the semantically understandable relational database patterns, in which it will allow the Conversational Agent and the Information Extraction Module to extract these patterns from conversations.

5.1.4 SQL Queries Templates

Using the SQL select statement structure to create a generalized set of SQL select statement templates. These templates will be used by the information extraction module in order to generate SQL select statements from the semantically understandable relational database patterns.

The presentation of the templates here is based on the complexity of the SQL select statement which is going to be generated.

This category include templates can create an SQL select statements based on one table.

```
{table name} | {table name & Column/s name} | {table name & Column/s value}
{table name & Column/s name & column/s value}
```

This category include templates can create SQL statements based on two tables.

{2 tables names} | {2 table's names & Column/s names} | {2 table's names & Column/s values} | {2 table's names & Column/s names & Column/s values}

This category include templates can create SQL statements based on three tables

{3 tables names} | {3 table's names & Column/s names} | {3 table's names & Column/s values} | {2 table's names & Column/s names & Column/s values}

5.1.5 Tree for Dynamically Generated Queries

The Tree for dynamically generated queries will serve the purpose of saving automatically generated SQL queries from the Conversation-Based Interface to Relational Database. The main purpose is saving dynamically generated SQL statements in a structured manner, since the SQL and associated conversation could be used to grow the static knowledge tree built by the knowledge engineer.

5.2 The Back End Level

This level contains the main knowledge repositories; it mainly contains two components. The first component is the Relational Database Meta Data and the Annotation Module. This information will be provided as part of the relational database management system such as the Meta Data and an Annotation Module. The second module is the Relational Database, which represent the dynamic knowledge base the system trying to answer queries from.

5.2.1 Relational Database Meta Data and Annotation Module

This module is mainly for annotating the database table's names, column's names and function's names, to be used in the Information Extraction Module.

5.2.2 The Relational Database

The relational database is an independent component since the proposed architecture can work with any relational database management system to obtain a result set for any SQL statement. The domain relational database used is for a global computing company. The Global Computing Company distributes computer hardware and software components to customers on a worldwide basis.

5.3 The Interactive Level

This level contains modules which work very closely with the user and are mainly represented by the Information Extraction, Response Generation, and the Conversational Manager.

5.3.1 Information Extraction Module

Information Extraction Module (IE) is responsible for extracting useful information from the user input in order to formulate SQL query. The extraction of information is carried out in two levels, the first level at the Conversational Agent (CA) level where the CA will extract information during the conversation; the CA will set out variables

i.e. pass through the nodes in the knowledge tree, these variables will be taken into account towards formulating SQL query. The second level is the dynamic mode level or when the conversation arrive to a point of not having enough information supplied to the conversational agent to generate a query from the trained knowledge i.e. knowledge tree extracted from the expert knowledge. Knowledge extracted at this level is based on all inputs have been entered by the user, the inputs will be sent to the Information Extraction Module this module will try to extract **Semantically Understandable Relational Database Patterns**, the semantically understandable patterns will formulate the basic building blocks of general SQL templates in order to dynamically generate the nearest SQL query from the user text.

Information Extraction techniques have been used: Text Cleaning, Semantic Interpretation, and Template Filling. In the Text Cleaning, the system removes duplicated words, and unnecessary tokens e.g. spaces. What to remove from the text as unnecessary tokens has been arguable. The system will keep all the words entered because any piece of text could be useful at some stage. Through the Semantically Understandable Relational Database Patterns, the system will extract and interpret table's names, column names, column values, and aggregation functions names. At last the system will try to fill the extracted information into the SQL templates available to formulate SQL statement.

5.3.2 Response Generation

The framework uses the response generation module in order to create responses to be sent to the user. There are six types of responses; first is the Initial Response which will come at the start of each context such as a welcome message or context identification. This response will give the user an idea where he is in the conversation and offer options for directions. The second is the In-Context Conversational Response, which is a context related dialogue response. This response allows further discussion within a context. The third is the Enquiry Response, which asks the user for further information in order to formulate an SQL statement. This response can appear when the conversation arrives to a leaf node i.e. goal node and when more information is needed to create the SQL statement. The fourth is the trained SQL statement and Result Set Response. This type of response comes directly from the knowledge tree when the user arrives to a leaf node and no further information is needed i.e. the goal has been reached. The fifth is the Dynamic enquiry response: once in the dynamic mode, the user will be asked for further information. The last type is the Dynamic SQL Statement and its Result Set Response. This response will come from the enhanced framework as a result of further processing of the inputted text.

5.3.3 Conversation Manager

The Conversational Manager in this framework is involved to manage more components than in the initial framework [1]. In this framework the goal is not known and the system has to create an SQL query dynamically through the use of the Conversational Agent, and Information Extraction in case the answer or the SQL statement is not in the trained knowledge tree. The Conversational Manager will manage all the modules in the interactive level in order to ensure the system flexibility.

6 Results

Results on the initial prototype and conversations shown in [1], in this paper the enhanced prototype has been developed and testing the prototype shown satisfactory results apart from the need to improve the interactivity and continuation of conversation between the user and the system after generating the dynamic SQL statement based on Information Extraction. Figure 3 is an example on the use of Information Extraction to project simple Natural Language Query into an SQL Statement. The highlighted keywords have been used in the Information Extraction module in order to fill an SQL template.

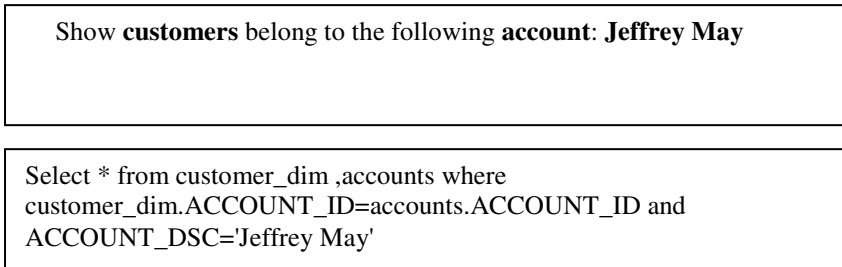


Fig. 3. Projection from natural language into SQL Statement using Information Extraction (IE)

7 Summary

The proposed enhanced framework introduces an easy and flexible way in order to develop Conversation-Based Interfaces to Relational Databases (C-BIRD), through the use of goal oriented conversational agent and knowledge tree. A goal oriented conversational agent provides the natural language interface and helps disambiguate the user's queries, by utilizing dialogue interaction. Knowledge tree is easy to create, structure, update, revise, and maintain. Further more the knowledge tree will direct the conversational agent towards the goal (i.e. query generation from natural language). The enhanced framework utilizes Information Extraction in order to create dynamic SQL statements from natural language sentences, by extracting semantically understandable relational database patterns and filling them into SQL templates in which they will make up answers to users queries. Using Information Extraction the prototype is now equipped with further capability to generate SQL statements dynamically from natural language statements, which will increase the accuracy of answering a user query.

References

1. Owda, M., Bandar, Z., Crockett, K.: Conversation-Based Natural Language Interface to Relational Databases. In: The 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, California, pp. 363–367 (2007)
2. Turing, A.: Computing machinery and intelligence. In: Computation & Intelligence: collected readings, pp. 23–46. American Association for Artificial Intelligence, Menlo Park (1950)

3. Reis, P., Mamede, N., Matias, J.: Edite – A Natural Language Interface to Databases: a New Dimension for an Old Approach. In: Proceeding of the Fourth International Conference on Information and Communication Technology in Tourism, Edinburgh, Scotland (1997)
4. Popescu, A., Etzioni, O., Kautz, H.: Towards a theory of natural language interfaces to databases. In: Proceedings of the 8th International Conference on Intelligent user Interfaces, January 12-15, pp. 327–327. ACM Press, Miami (2003)
5. Androutsopoulos, I., Ritchie, G., Thanisch, P.: MASQUE/SQL – An Efficient and Portable Natural Language Query Interface for Relational Databases. In: Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 327–330. Gordon & Breach Science Publishers, Edinburgh (1993)
6. Prasad, R., Ranjan, K., Sinha, A.: AMRAPALIKA: An expert system for the diagnosis of pests, diseases, and disorders in Indian mango. *Knowl.-Based Syst.* 19(1), 9–21 (2006)
7. Hendrix, G., Sacerdoti, E., Sagalowicz, D., Slocum, J.: Developing a Natural Language Interface to Complex Data. *ACM Trans. Database Syst.*, 105–147 (1978)
8. Woods, W., Kaplan, R., Webber, B.: The Lunar Sciences Natural Language Information System, Technical Report 2378, Bolt Beranek and Newman Inc. (1972)
9. Colby, K.: Simulation of Belief systems. *Computer Models of Thought and Language*, 251–286 (1973)
10. Introduction to ESTA (Expert System Shell for Text Animation), Prolog Development Center, Denmark,
<http://www.visual-prolog.com/vip6/Community/esta/esta.htm>
11. Sammut, C.: Managing Context in a Conversational Agent. *Electron. Trans. Artif. Intell.* 5, 189–202 (2001)
12. Yager, R.: Knowledge trees and protoforms in question-answering systems: Special Topic Section on Soft Approaches to Information Retrieval and Information Access on the Web. *J. Am. Soc. Inf. Sci. Techn.* 57, 550–563 (2006)
13. Appelt, E., Israel, D.: Introduction to Information Extraction. Tutorial at IJCAI 1999 (1999)
14. Wallace, R.: The Anatomy of A.L.I.C.E., Artificial Intelligence Foundation Inc. (2004),
<http://www.alicebot.org/anatomy.html>
15. Adam: Student Debt Advisor, Convagent Ltd. Manchester(2001),
<http://www.convagent.com/convagent/adam3.aspx>
16. Weizenbaum, J.: ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM* 9(1), 36–45 (1966)
17. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: Toward conversational human-computer interaction. *AI Mag. American Association for Artificial Intelligence* 22, 27–37 (2001)
18. Latham, A., Crockett, K., McLean, D., Edmonds, B., O’Shea, K.: Oscar: An intelligent conversational agent tutor to estimate learning styles. In: IEEE International Conference on Fuzzy Systems, 978-1-4244-6919-2 (2010)
19. Minock, M.: C-Phrase: A System for Building Robust Natural Language Interfaces to Databases. *Journal of Data Engineering (DKE)* 69(3), 290–302 (2010)
20. Kaetzel, L., Clifton, J., Klieger, P., Snyder, K.: Highway concrete (HWYCON) expert system user reference and enhancement guide. National Institute of Standards and Technology BFRL, Gaithersburg (1993)
21. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interfaces to databases-an introduction. *Journal of Language Engineering* 1(1), 29–81 (1995)
22. Johnson, T.: Natural Language Computing: The Commercial Applications. *Knowledge Engineering Review* 1, 11–23 (1986)

Multi Agent System for Historical Information Retrieval from Online Social Networks

Ruqayya Abdulrahman, Daniel Neagu, and D.R.W. Holton

Department of Computing, University of Bradford, Bradford, UK
{rshabdul, d.neagu, d.r.w.holton}@bradford.ac.uk

Abstract. Nowadays, we witness a flood of continuously changing information from a variety of web sources. The desire to track the information changes in real time requires new methods in web information retrieval. This paper explores the process of retrieving information from Online Social Networks (OSNs) using Multi Agent System (MAS) technology. The novelty of this study consists in leaving an agent in each user profile to monitor its updates which are sent to a controller agent that saves a history of each user's activity in a local repository. An algorithm making use of MAS within the Online Social Network Retrieval System (OSNRS) is proposed. Our experimental work shows that using MAS simplifies the process of tracking profile's history and opens the opportunity of understanding the dynamic behavior of OSN users especially when it is combined with text mining.

Keywords: agent, multi agent system, information retrieval, online social network.

1 Introduction

Retrieving information from different web resources such as websites, online databases and services is the area that information extraction research has recently concentrated on. However, to get valuable information that satisfies the user or application requirements, more efforts are needed to overcome the difficulties that are related to either the information representation or extraction methods. The fact that this data is represented in different formats such as text, video and audio, in addition to the rapid changes in their contents and templates, adds extra challenges to this research.

Lenhart and Madden [1] defined Online Social Networks as the “*online place where a user can create a profile and build a personal network that connects him or her to other users*”. Online Social Networks (OSNs) have been used widely by users of various ages. The increased simplicity in accessing the WWW via wireless devices such as laptops and smart phones helps end users to participate in OSNs. This is because it gives them opportunity to make new friendships, share their interests even with unknown people, upload photos and distribute their personal information. These changes in the users' profiles affect their networks' behaviors and actions and lead to alterations in the pattern analysis. Thus, more attention is required on how information will be collected from OSN websites taking into consideration these rapid changes [2] [3].

Our research focuses on retrieving information from OSN sites. MySpace¹, Facebook² and Twitter³ are on the top of the OSNs [4] containing vast amount of information produced on a daily basis. For example, currently, the time spent by UK web users on OSNs is about 23% of the total time spent on the web, compared with 9% in 2007 [5].

Although there are numerous studies that have attempted to extract millions of profiles from different OSNs, e.g.[5] [6]. To date, they have analyzed the results regarding visiting each profile only once. To the best of our knowledge, there is no reported work that deals with the monitoring of the rapid changes in those profiles.

This paper provides a method to investigate situations which require a continuous observation of the user's profile in order to track the changes that could help in understanding the structure of OSNs and its effects on different disciplines. For example, security personnel, i.e. police and criminal investigators, are interested to monitor suspicious profiles or web sites. Parents, guardians and social workers may also need to watch their children's online presence for safety reasons.

The aim of this paper is to improve our previous algorithm presented in [3] by using Multi Agent System (MAS) to overcome some of the limitations in preserving and recording the temporal ordering of OSN profiles' events, to speed up the extraction process and to make the algorithm more powerful.

The rest of the paper is arranged as follows: Section 2 highlights a brief overview on different approaches of information extraction from OSN. Our contributions to OSNRS and its organizational model and algorithm are described in detail in Section 3. Section 4 describes our experimental work on running some agents on MySpace profiles. The findings and initial results are analyzed in Section 5. Finally, in Section 6 the conclusion and future work is presented.

2 Background and Related Work

2.1 Online Social Networks Information Extraction Approaches

OSN sites have been analyzed by researchers using various extraction techniques. Most of the early research relied on non automated approaches to extract information through conducting surveys then analyzing the results. For example, [7] had interviewed security experts and conducted surveys of students to emphasize that they unknowingly played an important role in posing themselves and their campus networks to malicious attacks. [9-11] also used interviews and surveys to collect their information.

Since extracting information manually is time consuming and could not match the huge number of networks' profiles, automated information extraction is considered to be the saviour. Viswanath et al. [8] studied the evolution of the activity of more than 60,000 Facebook social network users over 2 years. They used crawlers to get the interactions between all friends of a given user profile and friends of their friends in a Breadth First Search approach. From over 800,000 logged interactions between users,

¹ <http://www.myspace.com>

² <http://www.facebook.com>

³ <http://www.twitter.com>

they found out that the majority of those interactions are generated by a minority of user pairs. Bonneau et al. [12] retrieved information using crawlers, but for privacy issues. They introduced three different methods for information extraction. *Public listing*: to crawl public profiles, *false profiles*: to create a false profile to crawl “searchable” profiles, and *profile compromise and phishing*: which will target random or specific accounts using malicious applications and phishing attacks. They concluded that the existing privacy protection systems in OSNs are not consistent with users’ expectations.

2.2 Online Social Networks Information Extraction Using Agents

Various definitions of an agent are found. Wooldridge and Jennings in [13] define an agent as “*a computer system that is situated in some environment and capable of autonomous action in this environment in order to meet its design objectives*”.

A collection of agents that communicate and cooperate with each other to achieve a common goal are called a Multi Agent System (MAS). To the best of our knowledge, the research in retrieving information from OSNs using MAS technology is limited to the studies done by Chau in [14]. He extends previous work by Cho [15] but to work on a dynamic assignment architecture where a central coordinator (Master Agent) will control other crawler agents. They used Breadth First Fashion to crawl the eBay network and list the pending users who are seen but not crawled yet in a queue data structure. Each crawler agent sends a request to the Master Agent for the next user in the queue to be crawled. Then it uses multiple threads for crawling and returns the extracted information to the Master Agent. The Master Agent ensures that crawlers are working in parallel and no redundant crawling occurs. The work presented in this paper is similar to [14]. However, our agents are designed to stay in each profile to keep listening for updates as we will illustrate in the next sections.

3 Methodology

In order to appropriately address our methodology, we present an overview of our contributions to Online Social Network Retrieval System. Then we illustrate the organizational model and the algorithm of the proposed system.

3.1 Contributions to Online Social Network Retrieval System

Our proposed Online Social Network Retrieval System (OSNRS) is a MAS that consists of a finite set of agents (grabber agents) controlled by a special agent called MasterAgent. Through this paper, the grabber agent (described by *gAg*) is used to refer to the agent that is responsible for retrieving information from OSN profiles. In contrast, the MasterAgent (which is described by *mAg*) organizes the retrieving process and saving information in the local repository. More details will be provided in Sections 3.2 and 3.3.

OSNRS aims to retrieve information from a given OSN profile (seed URL) in a multi-threading approach. Each *gAg* will be allocated to a unique URL address and will remain there in order to keep listening to any updates in the profile.

The retrieved information will be sent back to the *mAg* using different port. To speed up the extraction process, the port is set up just for transferring files to distinguish it from the port that is used by agents for communication with the *mAg*. The *mAg* will then create a history of each profile, which is recorded in a local repository to be analyzed and mined later.

What is novel in this OSNRS is that, the *gAg* will stay in the assigned profile and keep listening to any updates or changes in that profile. The updates are detected by comparing the current retrieved information with the saved file of the previous extraction process. Once a change is captured, the *gAg* will inform the *mAg* and send a copy of the new changes.

MAS has several features that fit well with our OSNRS. The autonomy of *gAg* allows it to operate as a standalone process to achieve its goal in retrieving information without direct intervention of users. The perceptivity of *gAg* permits it to detect the changes in the profiles and to make decisions when it has to report the results to *mAg*. The mobility feature permits the *gAg* to convey between different machines in the network. The parallelism of *gAg* helps to speed up the operation of extraction. The scalability of MAS will facilitate the process of adding new *gAg* or other agents for different proposes as required without having to change the existing system. The sociability allows *gAg*, *mAg* and users of OSNRS to communicate with each other and exchange their knowledge.

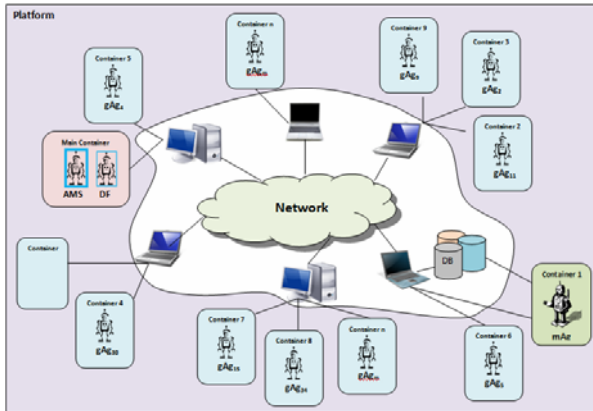


Fig. 1. Structural Model of OSNRS

3.2 Organizational Model of OSNRS

Java Agent Development framework (JADE) is one of the most significant open source software for designing MAS in recent years. As [16,17] JADE has been selected to implement OSNRS. In JADE, each agent should live in a running instance of runtime environment that is called a container. A composite of containers comprises the platform. Although many agents could live in one container, and several containers could compose several platforms, we decided to simplify the OSNRS environment by:

- Assigning one platform to compose all containers.
- Creating each agent in a unique container.

Fig. 1 shows the structure of OSNRS and its environment. Note that the number of containers in a workstation could be different. Also, the local repository (Database) could be accessed just by the *mAg*.

3.3 Algorithm of OSNRS

Recent developments in the field of MAS have led to increase researchers' interest in adopting it to be used in various domains including OSN.

As explained in the introduction, the objective of the proposed algorithm is to overcome the limitations in our previous approach [3] by applying MAS technology. The improved algorithm intends speeding up the extraction process, distributing

```

Given:
- Set the platform.
- Set the containers.
- Set the MasterAgent.

Input:
- Enter the required profile URL (seed URL).
- Choose one of the stopping criteria:
  o Number of friends.
  o Iterations level.
- Enter number of friends or number of iteration level.
- Set time to activate the agent.

Output:
  A historical personal information and list of friends for
  all public profiles starting from the seed URL.

Steps:
1) Send the URL address of the profile to the MasterAgent.
2) MasterAgent will check stopping criteria.
3) While not match the specified number to stop
   Begin
   a. Look for available grabber agent in the platform if
   any. Otherwise, create/clone a grabber agent.
   b. Allocate the agent to the URL through a message.
   c. Call grabInfo(seed URL).
   d. Receive the file and save it in a database.
   e. Get the list of friends and then go back to step2 using
   Breadth first search approach.
   End
4) Re-activate all grabber agents at the specified time

grabInfo (seed URL)
{
  The grabber agent will:
  1. Validate the web page.
  2. Retrieve the personal information and list of friends
  using our pervious method (Alim, 2009).
  3. Save a copy of the retrieved information in a file.
  4. If the profile has been browsed before, then compare
  the current information with the existing file.
  5. Attach a copy of the file with the message to the
  MasterAgent through different port to speed up the
  extraction process.
  6. Keep listening to detect any updates.
}

```

Fig. 2. The Algorithm of the Proposed OSNRS

agents in a multi-threading approach using Breadth First Search algorithms, skipping non-extractable URLs without crashing the system, and keeps listening to updates.

Our algorithm for OSNRS (illustrated in Fig 2) outlines the approach for retrieving profile's personal information and a list of top friends from the MySpace social network. MySpace has been chosen to be the domain of OSNRS on the basis of its allowance for non registered users to access public profiles easily.

As mentioned before, the main drive of our contribution is to keep an agent in each visited profile to track the updates and send a message to the *mAg* containing the new changes. Through this process, historical information of every profile will be saved in a local repository for future analysis and mining. In the sections that follow, the experimental work of applying this algorithm will be explained followed by findings and results.

4 Experimental Work

In our experiment, we did the following:

1) *Setting the environment of OSNRS*: Two workstations were connected to share one JADE platform. The containers are distributed on the two workstations to form the runtime environment. Initially, the *mAg* is created and connected to the database. PostgreSQL⁴ is chosen to save the retrieved information.

2) *Choosing the sample of OSNRS*: MySpace OSN is selected to be the domain of our experiment. Two groups of sample have been chosen as follow: for the first group, 20 random public profiles are set to be the seed URLs. There are no connections between these profiles, i.e. each profile is not in the list of top friends of other profiles. The second group was formed by choosing the profile that has the largest number of top friends, in our case 35 top friends. Subsequently, each profile in this list was established as a seed URL for extraction. As a result, we ensured that there are connections between these profiles.

3) *Running the OSNRS*: To run the system, we send each seed URL from both groups to the *mAg*. The *mAg* allocates a *gAg* to the seed URL to retrieve the list of top friends and number of all friends in addition to the personal information such as names (first name, last name and nick name), gender, age, country and link to photo. The *gAg* sends back this information to the *mAg*. Then, the *mAg* takes the list of top friends and allocates a unique *gAg* to each URL. The process is repeated as described in the methodology. In this stage of research, we are concerned about the top friends of each profile rather than all friends, and will stop if the specified number of friends is accomplished or if the second iteration is met; i.e. the friends of the friends of the seed profile. The retrieved information is not subject to rapid changes, therefore we decided to re-activate (awake) the agent once a day for two weeks to detect new updates.

5 Findings and Results

Since the target of this research focuses on justifying the proposed algorithm rather than being concerned about the profile contents, our experiment's simple goal was to

⁴ <http://www.postgresql.org/>

allow the agent to track the updates in the profile. The goal is set to see how often the users change their friends either in the total number of friends or in their top list of close friends. To evaluate the accuracy of the retrieved results, we tested some profiles manually, particularly in cases that reported unexpected results as will be shown in the rest of this section, e.g. the empty cells in the profile no. 3 in tables 2 and 3. The results show that all retrieved information is correct. However, future work to extend *mAg* functions to randomly check the contents of information retrieved at random times is ongoing.

Table 1. Comparison between Group 1 and Group 2

Features	Group 1				Group 2			
	TopFriends		AllFriends		TopFriends		AllFriends	
	Profiles	%	Profiles	%	Profiles	%	Profiles	%
Total	84				59			
Min no. of Friends	1		3		3		9	
Max no. of Friends	39		535968		37		2150503	
Average of Friends	10.9		15190.64		12.55		54262.07	
Mode of Friends	6		20		7		43	
Changed	1	1.19	20	24	1	2	19	42.22
Unchanged	60	71.43	37	44	35	78	17	37.78
IncompleteChanged	2	2.38	15	18	0	0	5	11.11
IncompleteUnchanged	18	21.43	11	13	9	20	4	8.89

Although OSNRS is a work in progress, the initial results were promising. The number of public profiles of both groups ranged between 130 and 160 according to the change in the sub network. Almost 80% of the visited profiles were aged below 30. Just over half of them (56%) were female. Table 1 presents results obtained from the preliminary analysis of the retrieved information. It shows that the user profile has a significant difference between the minimum and maximum number of all friends.

The mode of the number of top friends in both groups was similar while it was double regarding the number of all friends. We used NodeXL⁵ open source application to illustrate the friend networks of the profiles.

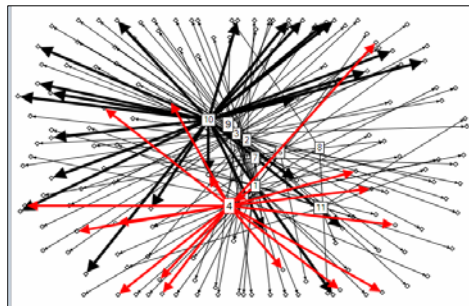


Fig. 3. Sub network of a profile in week 1

⁵ <http://nodexl.codeplex.com/>

Table 3. Sample of tracking number of all friends for 14 days

Profiles	No. of All Friends													
	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
1	9	9	9	11	12	12	12	12	12	12	12	12	12	12
2	34	34	34	34	34	34	34	34	34	34	34	34	34	34
3	154			154	154	154	154	154	154	154	154	154	154	154
4	312	312	312	312	312	312	312	312	312	312	312	312	312	312
5	692	692	692	692	692	692	692	692	692	692	692	692	692	692
6	1165	1165	1165	1165	1165	1165	1165	1165	1165	1165	1165	1165	1165	1165
7	7595	7594	7593	7592	7591	7590	7589							
8	17487	17484	17476	17472	17465	17456	17451	17448	17581	17575	17568	17566	17565	
9					20808	20808	20806	20806	20806	20805	20803	20801	20800	20797
10	51888	51879	51871	51869	51862	51862	51852	51848	51837	51832	51828	51824		
11	65440	65427	65408	65399	65370	65347	65329	65315	65299	65276	65260	65249	65242	65230
12	459724	459610	459406	459321	459059	458788	461503	461402	461209	461611	461564	461430	461324	

dropped from the list of top friends of someone's profiles. Two days later, it has appeared again in the list of top friends of another profile. The fact that the profile already exists in the database will undoubtedly help in understanding the impact of this profile on the sub network.

Also, we can conclude that the profile's large number of friends in the top list does not imply that all the profiles are retrieved. This depends on how many of those profiles are public. Moreover, missing information due to privacy issues will need to be factored into studies that analyze information from OSNs.

6 Conclusions and Future Work

This paper investigates the retrieval of information from OSNs through MAS technology implementation of OSNRS. OSNRS is designed with multi-threaded agents in which each agent's thread is assigned to a specific profile to track information changes in the users' profiles. This study will serve as a base for future studies in the process of tracking user profile's history and understanding the behavior of OSN users especially when combined with text mining.

Our process to correctly and accurately monitor and retrieve profile information is not affected by the size of the retrieved information. Nevertheless, OSNRS will be further implemented for the extraction of information from a large sample and over a longer period of time than that presented in this paper. That will help provide a better understanding of the behavior of the network. However, in order to achieve that, the implementation of OSNRS will be modified to enable extracting of the profile's information using Application Programming Interface (API), in order to be aligned with recent changes to the structure of social network developing methods. Also further work is required for evaluating *mAg* functions to check the contents of the retrieved information.

References

1. Lenhart, A., Madden, M.: Social networking websites and teens: An overview (2007), http://www.pewinternet.org/~media/Files/Reports/2007/PIP_SNS_Data_Memo_Jan_2007.pdf

2. Ye, S., Lang, J., Wu, F.: Crawling Online Social Graphs. In: The 12th International Asia-Pacific Web Conference (APWEB), pp. 236–242 (2010)
3. Alim, S., AbdulRahman, R., Neagu, D., Ridley, M.: Data retrieval from online social network profiles for social engineering applications. In: International Conference for Internet Technology and Secured Transactions, pp. 1–5. IEEE, Los Alamitos (2009)
4. Top 10 Social Networking Websites & Forums (November 2010), <http://www.marketingcharts.com/interactive/top-10-social-networking-websites-forums-november-2010-15359/> (last accessed on 11-01-2011)
5. Caverlee, J., Webb, S.: A large-scale study of MySpace: Observations and implications for online social networks. *Artificial Intelligence* 8, 36–44 (2008)
6. Mislove, A., Marcon, M., Gummadi, K., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: The 7th ACM SIGCOMM Conference on Internet Measurement, pp. 29–42. ACM, New York (2007)
7. Gibson, R.: Who’s really in your top 8: network security in the age of social networking. In: The 35th Annual ACM SIGUCCS Fall Conference, pp. 131–134. ACM, New York (2007)
8. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.: On the evolution of user interaction in facebook. In: The 2nd ACM Workshop on Online Social Networks, pp. 37–42. ACM, New York (2009)
9. Acquisti, A., Gross, R.: Imagined communities: Awareness, information sharing, and privacy on the Facebook. In: *Privacy Enhancing Technologies*, pp. 36–58. Springer, Heidelberg (2006)
10. Dwyer, C., Hiltz, S., Passerini, K.: Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. *Information Systems Journal*, 13 (2007)
11. Strater, K., Richter, H.: Examining privacy and disclosure in a social networking community. In: *Proceedings of the 3rd Symposium on Usable Privacy and Security*, pp. 157–158. ACM, New York (2007)
12. Bonneau, J., Anderson, J., Danezis, G.: Prying data out of a social network. In: *The International Conference on Advances in Social Network Analysis and Mining*, pp. 249–254. IEEE, Los Alamitos (2009)
13. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. *The Knowledge Engineering Review* 10(2), 115–152 (1995)
14. Chau, D., Pandit, S., Wang, S., Faloutsos, C.: Parallel crawling for online social networks. In: *The 16th International Conference on World Wide Web*, pp. 1283–1284. ACM, New York (2007)
15. Cho, J., Garcia-Molina, H.: Parallel crawlers. In: *Proceedings of the 11th International Conference on World Wide Web*, pp. 124–135. ACM, New York (2002)
16. Camacho, D., Aler, R., Castro, C., Molina, J.: Performance evaluation of zeus, jade, and skeletonagent frameworks. In: *IEEE International Conference on Systems, Man and Cybernetics*, p. 6. IEEE, Los Alamitos (2003)
17. Aldea, A., Banares-Alcantara, R., Jimenez, L., Moreno, A., Martinez, J., Riano, D.: The scope of application of multi-agent systems in the process industry: three case studies. *Expert Systems with Applications* 26(1), 39–47 (2004)

Social-Network Influence on Telecommunication Customer Attrition

Piotr Wojewnik, Bogumil Kaminski, Mateusz Zawisza, and Marek Antosiewicz

Decision Support and Analysis Division,
Warsaw School of Economics, Al. Niepodleglosci 162, Warsaw, Poland
bkamins@sgh.waw.pl,
{piotr.wojewnik,mat.zawisza,marekantosiewicz}@gmail.com

Abstract. We investigate the network effects of churn in the telecommunication industry. Under calling party pays regime differentiation between on-net and off-net prices implies that customer's calling cost depends on operators chosen by the clients he calls. We assume that clients minimize their expenses. Therefore, after a single person churn we observe churn induced in his social neighborhood. Our aim is to verify, which measures of individual position in a social network are important predictors of induced churn. We control the results for changes in market prices structure, social network structure and number of operators on the market. Using multiagent simulation we show that (a) network structure and number of operators significantly influence induced churn level and (b) weighted prestige is its important predictor.

Keywords: Telecommunication, Customer attrition, Agent-based Models, Social-network analysis.

1 Introduction

One of the key issues driving telecommunication operator's decision making is the ability to forestall losing customers to competition. This is motivated by the fact that acquiring new customers is between 5 to 8 times more expensive than retaining existing ones [10,21]. The level of such voluntary attrition (churn) is estimated as 25-50% of client base per annum, where the particular rate depends mainly on client's segment [4]. The problem of voluntary churn is especially viable for pre-paid customers, where the client might cancel the contract without consequences. The attrition ratio can be reduced by retention campaigns. In order to optimize such actions operators stratify their clients according to their propensity to churn. The process of churn prediction model creation should be automated, because applications reveal the problem of model ageing [3].

Major reasons for voluntary attrition, identified in the literature, are: unsatisfactory service level, unattractive handset portfolio, uncompetitive pricing, poor coverage and bad experiences with call centres [6,11]. This is reflected in standard analytical models forecasting customer churn, where the following predictor domains are used: customer personal socio-economic characteristics, trends

in number and length of calls over a long period [18], customer's dissatisfaction level measured as number of calls to the mobile carrier's call center [5]. Some modelers employ also minutes of use, balance forward from previous monthly bills, and the tenure of the customer [3]. Generally authors of churn models start with hundreds of variables and choose the most important with the help of some feature selection methods. In recent years clients are analyzed as a social network. Hence, the list of churn predictors includes also some simple network features like connectivity (e.g. number of neighbors) and interconnectivity (e.g. number of churn neighbors connected by an edge) [2].

In our research we focus on customer attrition under *caller party pays* market model. If the charges for the on-net and off-net traffic differ, the optimal choice of operator depends on prices themselves and individual's call structure. Therefore, when a client churns, it might raise the propensity to churn of people calling him, because their call structure changes. We will call such behavior *an induced churn*. Consequently, the problem of identification and monitoring of clients whose churn might lead to high induced churn is important for telecommunication operators.

The objective of this paper is to identify *universal predictors* of the influence of individual customer churn on attrition of other customers. This aim is motivated by the following business scenario. A customer churns due to non-price reasons (for example: poor handset offer, bad experience with customer service). Telecommunication operator wants to estimate how many of his other customers might also churn due to change in their calling costs (induced churn).

We aim to identify measures of customer's position in calls network that are best predictors of induced churn under varying (a) market price structure, (b) call network structure and (c) number of operators. The recognized characteristics will allow telecommunication operators to better identify network users with highest impact on the attrition propensity in their client base. In order to achieve this objective we construct a multiagent model of telecommunication market and simulate churn behavior of customers under different structural assumptions. This differs from standard literature approach, where reported churn prediction models use a single data set, which limits the power of the obtained results to a single market only.

2 Model Architecture

The model consists of two types of agents (1) mobile network subscribers (exactly one mobile phone number) and (2) mobile network operators.

There are n mobile subscribers, who are characterized by the following factors: the operator they are using, a list of unique clients they are calling along with calls' intensity during a certain period of time (e.g. a month). Based on this information, the subscribers are placed in a *calls network* - a weighted directed graph, where the clients are represented by nodes and the intensity of their calls are weights of edges.

Each subscriber i has the ability to calculate his total calling cost for his current operator as well as his potential total calling cost for all other operators. The total calling cost takes into account call frequency (represented as matrix $D = [d_{ij}] \in \mathbb{R}_+^{n \times n}$) and prices of operator k ($c_{k,1}$ for on-net and $c_{k,2}$ for off-net prices). The cost assuming client i chooses operator k is the sum of his on-net and off-net calls volume, multiplied by on-net and off-net prices:

$$\text{cost}_i(k) = \sum_{\{j:\text{operator}(j)=k\}} d_{ij}c_{k,1} + \sum_{\{j:\text{operator}(j)\neq k\}} d_{ij}c_{k,2}$$

where function $\text{operator}(j)$ returns the operator number of j^{th} subscriber. Subscribers are aware of network operators being used by their peers at any moment.

The assumption, that the subscribers know the network operators used by their peers, seem to be strong, though e.g. in Polish it is plausible. In particular, in 2010 the four main Polish mobile operators possessed over 95% of the market, and their prefixes for new clients are generally known. These clients might churn carrying their phone number with, whereas the level of such migration is only of 2-3% in a year, so the prefix is very good predictor as to the operator used. Moreover, the subscribers might recognize the operators of their peers basing on the billing list (in postpaid market) or single call cost (in prepaid market).

In each time period of simulation, subscribers are activated in a random order. Upon activation, subscribers optimize the choice of their operator. In order to do this, he calculates his total calling cost for each operator and chooses an operator with the lowest total cost.

The *operator* agent is characterized by his calls tariff: the pair of prices for calling to subscribers in the same network (on-net) and in another network (off-net). Since we consider only the short term market effects, we assume operators keep their tariffs constant during a single simulation. Operators do not perform any optimization of their tariffs, so the on-net and off-net prices are modeled as external parameters.

The *calls network* is modeled as a Small World Network, because such topology approximates call graphs for the whole populations well, see case of Finland [8,14,15]. It is generated by the procedure proposed by [20], which is parametrized by *rewiring probability* μ and *radius* r . Firstly, we create one-dimensional lattice network with radius r . On the lattice network, nodes are equally distributed on a circumference, and each node is connected to his nearest neighborhood within radii of r . Hence, each node has $2r$ connections (neighbors on the left and right), which is average degree of Small World Network [16]. Lattice network features high level of clustering, which is typical for telecom networks. Next this lattice network is altered randomly. For each node we divide the links into two groups: left and right. Next we replace the right links with new remote connections with rewiring probability at level μ . Higher value of parameter μ makes the resulting social network more similar to random network than lattice network. In extreme case the calls graph has random network property - the existence of each link is equally probable. Hence, the rewired network is always between lattice network

($\mu = 0$) and random network ($\mu = 1$). Weights of network links are drawn from the exponential distribution, which is parametrized by average call frequency $\lambda = 100$.

Network parameters μ and r are chosen to fit real data taken from anonymous telecommunication operator. Using theoretical probability distribution according to [1], we optimize μ and r to minimize sum square difference between theoretical and empirical probability distribution of neighborhood size. The following objective function is taken:

$$\text{objective}(\mu, r) = \sum_{i=0}^n (\widehat{f_{\mu,r}}(i) - f_{emp}(i))^2 \rightarrow \min$$

where $\widehat{f_{\mu,r}}(i)$ is the theoretical frequency of clients, having exactly i neighbors, for the network with μ and r parameters. This is compared to empirical frequency $f_{emp}(i)$ from the real data. The estimated parameter values resulting from this procedure are $\hat{\mu} = 1.00$ and $\hat{r} = 3$.

Moreover, we depict the frequency of neighborhood size on Figure 1, as well as, present the descriptive statistics in Table 1. We can observe that the average number of neighborhoods is 7.475, which corresponds to radius r of 3 or 4, since mean neighborhood size in Small World Network is $2r$.

Table 1. Descriptive statistics of neighborhood size.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max	St. Dev.
1	3	5	7.475	9	69	7.83

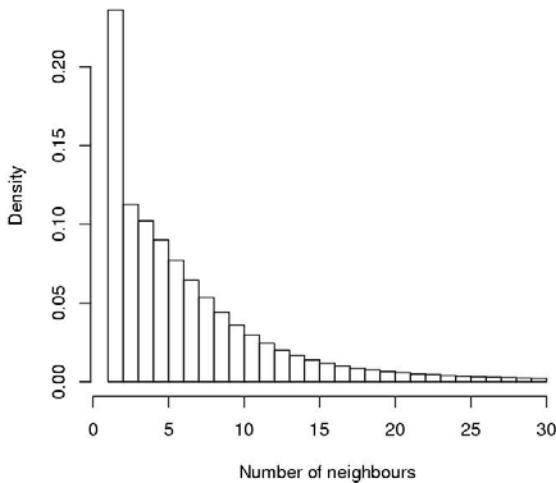


Fig. 1. Histogram of neighborhood size

Following the objectives of the paper we do not want to stick to single network parametrization and we use a range of μ and r parameters values. Such approach allows us to obtain generalizable results and check their robustness.

3 Simulation Results

3.1 Experiment Setup

The parameters of the simulation model are given in Table 2. In our experiment we explore 54 different parameter configurations. For each parameter setting we run the simulation $T = 4$ times. The parameters that are constant for each simulation are: the number of subscriber agents ($n = 1500$), and the parameter for the exponential distribution of the intensity of calls ($\lambda = 100$). The number of operators can be equal to 3 or 5. The price tariffs are defined as a pair of an on-net price and off-net price levels, which is the same for every operator. Normalizing the on-net price to unity, the off-net price can be equal to 1.1, 2 and 3. The experiments are also carried out for different parameter values for the procedure generating the call network. The rewiring probabilities used are 50%, 75% and 100%, and the radius takes the values of 3, 4 and 5. For each simulation a new call network is generated.

The first step of the simulation is to generate a Small World Network for 1500 agents according to the method described in Section 2. Next, the set of measures of customer position in the call network is calculated. The list is given in Table 3. The first three are simple measures of direct links between the subscriber i and its peers. *Degree* is the size of the neighborhood of subscriber i , where the neighborhood is defined as the set of peers who both call and are called by subscriber i . *Prestige* is the number of other subscribers who call subscriber i ,

Table 2. Model parameters and their values

Parameter	Simulation values	Meaning
Telecommunication Market		
n	1500	The number of subscribers in the system
k	{3, 5}	The number of operators
c	$\forall_i c_{i,1} = 1$ $\forall_i c_{i,2} \in \{1.1, 2, 3\}$	Price tariffs c . c is a $k \times 2$ matrix, where $c_{i,1}$ is on-net price and $c_{i,2}$ is off-net price of operator i
Telecommunication Network		
μ	{50%, 75%, 100%}	Rewiring probability
r	{3, 4, 5}	Radius
λ	100	Average call frequency
D	$\mathfrak{R}_+^{n \times n}$	Distance matrix of call network, containing the information of call time. d_{ij} is call time the caller i talks with the receiver j .
Simulation Technicality		
T	4	Number of times whole simulation is run for single parametrization

Table 3. Measures of subscriber's position in the call network

Measure	Symbol	Meaning
Degree	DEG	Number of other subscribers - called by and calling the subscriber [12]
Prestige	PRE	Number of other subscribers who call the subscriber [12]
Weighted prestige	WPR	Sum of intensities of other subscribers' calls to this subscriber [12]
Local clustering coefficient	LCC	Degree to which callers of the client consist of a clique [20]
Eigenvector centrality	ECT	Importance of subscriber [9,19]

and *Weighted Prestige* is the total sum of call intensities from other subscribers to subscriber i . The next two measures take into account the whole structure of the call graph and not just direct links between pairs of subscribers. *Local Clustering Coefficient* is the number of links between the subscribers who belong to the neighborhood of subscriber i divided by the total number of possible links, which is $s(s-1)$, where s is the size of the neighborhood. *Eigenvector Centrality* of subscriber i is the i -th coordinate of the eigenvector corresponding to the greatest eigenvalue of the call network matrix. It is the measure of subscriber's centrality within a network.

The experiment is performed in two phases (a) *burn in* period and (b) *analysis* period. In the burn in phase each subscriber is assigned randomly to a network operator and subscribers optimize their call costs by changing the network operators according to the optimization rule described in Section 2. The optimization phase is simulated for 1000 iterations, which is always enough for the operator choice of each agent to become stable.

Next, the analysis period is performed as follows. Starting from the network structure (the choice of operators) obtained after the burn in phase, the following procedure is applied for each subscriber. We switch the operator of a chosen subscriber to some other random operator and force him to remain with that operator. We will call this operation *forced churn* and interpret it as originated by reasons other than calling cost minimization (for instance customer's dissatisfaction with operators offer of handsets).

Afterwards we run the simulation for another 50 iterations with a probability $p = 0.1$ for each agent to optimize their operator choice in a single period. The customers that switch operators during this period are induced churners. We calculate two indices of the level of induced churn. Firstly, we calculate the total number of subscribers who have chosen a different network operator than the one chosen after the burn in period. This index takes into account the number of churners from the operator of the initial forced cherner as well as churners from other operators. Therefore we calculate a second measure of induced churn level, which is the number of churners from the initial operator of the forced cherner. After this, agents return to their operators chosen after the burn-in period and this step is repeated for next subscriber.

Using the above simulation procedure, for each customer, we obtain his metrics of social position in the call graph and two measures of his induced churn level. For a single parametrization the procedure is repeated T times. Simulations are conducted using MASON Multiagent Simulation Toolkit [13], as well as JUNG and JAMA packages of Java programming language. As a result we obtain a data set for each parametrization of the simulation ($3 * 3 * 3 * 2 = 54$ datasets containing $4 * 1500 = 6000$ observations).

To identify churn predictors we binarize induced churn values. Since for 6000 clients only a low proportion of induced churn measures is greater than 1, we create a binary variable $target = 0$ for *zero churn* and $target = 1$ for *non-zero churn*. Such formulation enables us to employ logistic regression to identify the factors influencing the churn intensity. We also tested an alternative – linear probability model [7] – to check the robustness of the logistic model. The results do not differ significantly.

The following network characteristics have correlation coefficient above 0.7: *prestige*, *weighted prestige* and *degree*. This fact makes its coefficient estimates, in logistic regression, negatively correlated. As a consequence, we abandon both *prestige* and *degree* variables, because their information is nearly contained in *weighted prestige*. In the logistic regression model we estimate the probability that he will induce churn from the following formula:

$$P(target = 1) = \frac{\exp(\beta_0 + \beta_1 WPR + \beta_2 LCC + \beta_3 ECT)}{1 + \exp(\beta_0 + \beta_1 WPR + \beta_2 LCC + \beta_3 ECT)},$$

where coefficients are computed from data by Maximum Likelihood Estimator [7]. We work on the standardized data, so that coefficients are comparable with each other and denote the strength of churn intensity of respective variables. Computational routines are conducted in GNU R [17].

3.2 Social Factors Influencing Customer Attrition

The results of performed simulations are given in Table 4 and Table 5.

What seems non-obvious, is the impact of average radius, i.e. average neighborhood size, on the average churning. As we can see in Table 4, the higher radius r , the smaller churning. This is explained by the fact, that in small neighborhoods, each neighbor has greater relative impact on a given client, so the induced churn occurs more often. On the other hand, in larger neighborhoods, the switching of single neighbor might be not sufficient to convince others to change an operator.

The existence of difference between on-net and off-net prices are necessary for churn to take place. Additional simulations were conducted where $c_{i,1} = c_{i,2}$ to confirm that result. The size of this price difference plays role only for the small neighborhood of radius $r = 3$. In Table 4, we see that larger difference coincide with higher churn for $r = 3$. However, this relationship does not hold for larger neighborhood size of $r = 4$ or $r = 5$.

We also identify the impact of the number of operators: the higher the number of operators, the more induced churn we observe. This result should encourage

Table 4. Average number of churned clients induced by subscriber’s switching. For each parametrization we present: the overall churning and a ratio of churners for the initially churned customer’s operator.

r	μ	$k = 3$			$k = 5$		
		$c_{i,2} = 1.1$	$c_{i,2} = 2.0$	$c_{i,2} = 3.0$	$c_{i,2} = 1.1$	$c_{i,2} = 2.0$	$c_{i,2} = 3.0$
3	50%	3.03(85%)	3.12(87%)	3.24(87%)	3.42(81%)	3.20(82%)	3.54(80%)
	75%	2.43(87%)	2.22(88%)	2.88(86%)	3.88(77%)	3.55(77%)	2.77(79%)
	100%	1.99(85%)	2.71(85%)	3.83(83%)	2.31(79%)	2.31(78%)	2.55(81%)
4	50%	1.10(86%)	1.22(81%)	0.17(100%)	1.11(73%)	1.02(79%)	0.20(100%)
	75%	0.18(100%)	0.16(100%)	0.18(100%)	0.17(100%)	0.16(100%)	0.16(100%)
	100%	0.18(99%)	0.16(100%)	0.19(100%)	0.17(100%)	0.17(100%)	0.15(100%)
5	50%	0.06(100%)	0.72(99%)	0.06(100%)	0.07(100%)	0.06(100%)	0.06(100%)
	75%	0.06(100%)	0.06(100%)	0.07(100%)	0.06(100%)	0.06(100%)	0.06(100%)
	100%	0.07(100%)	0.06(100%)	0.08(100%)	0.07(100%)	0.07(100%)	0.06(100%)

Table 5. Importance of subscribers’ network characteristics for different parametrizations. Each of 18 estimated logistic regressions contains coefficients for: weighted prestige, local clustering coef. and eigenvector centrality. The absolute value of each coefficient divided by the largest coefficient in the whole sample gives the normalized measure of variable importance.

r	μ	$c_{i,2}$	$k = 3$			$k = 5$		
			WPR	LCC	ECR	WPR	LCC	ECR
3	0.50	1.1	0.81	0.04	0.12	0.98	0.02	0.02
		2.0	0.84	0.03	0.06	0.94	0.01	0.02
		3.0	0.80	0.01	0.03	0.90	0.02	0.05
	0.75	1.1	0.75	0.01	0.10	1.00	0.01	0.02
		2.0	0.69	0.01	0.17	0.95	0.03	0.03
		3.0	0.73	0.01	0.06	0.74	0.05	0.00
	1.00	1.1	0.66	0.03	0.09	0.66	0.01	0.01
		2.0	0.71	0.01	0.38	0.64	0.01	0.04
		3.0	0.96	0.01	0.05	0.75	0.02	0.14

telecommunication regulators to enable new operators to enter the market and boost competition.

Table 4 also shows that the lower rewiring probability μ , the higher average churning rate. Low μ makes more cliques to occur in the network, which could contribute to the domino effect, i.e. one churned client will result in a few more churn events. Finally we note that the majority of induced churn events happen for the initially churned customer’s operator.

Next let us analyse the role of client’s social network position on induced churn, see Table 5. Variable *weighted prestige* has the greatest impact on induced churn. This holds for all network parametrizations and is statistically significant at the level of $\alpha = 1\%$. However, we observe higher importance of this variable for small clustered neighborhood, i.e. with $r = 3$ and $\mu \in \{0.5, 0.75\}$. *Eigenvector centrality* is found important just occasionally. We found no significant

impact of *local clustering coefficient*. The same calculations were performed for $r = \{4, 5\}$, resulting in similar conclusions.

4 Conclusions

The problem of monitoring induced churn is crucial for the maintenance of profitable customer retention policy by telecommunication operators. Next to the typical factors like: customer service satisfaction, prices or device availability, the customers choosing the operator consider also their individual call structure. Therefore its changes might increase churn level.

We have shown that churn induced by switching of a single customer is the most significant issue for markets with large number of operators and call networks with low radius. At individual customer level we have found that weighted prestige is a good predictor of induced churn independent of simulation parametrization.

The presented findings are important for shaping the telecommunication operators marketing policies. Using weighted prestige measure they can identify the clients whose churn can cause most induced churn. In such situation operator will be willing to invest into ensuring that such customers do not switch their provider.

In further research it is interesting to test a broader range of customer's network position characteristics, such as: betweenness or closeness and check robustness of our results through implementing different subscribers' operator choice rules.

Acknowledgments

This research was supported by Warsaw School of Economics Research Grant number 03/S/0025/10.

References

1. Barrat, A., Weigt, M.: On the properties of small-world network models. *The European Physical Journal B - Condensed Matter and Complex Systems* 13(3), 547–560 (2000)
2. Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A.A., Joshi, A.: Social ties and their relevance to churn in mobile telecom networks. In: *EDBT 2008: Proceedings of the 11th International Conference on Extending Database Technology*, New York, NY, USA, pp. 668–677 (2008)
3. Datta, P., Masand, B., Mani, D.R., Li, B.: Automated Cellular Modeling and Prediction on a Large Scale. *Artificial Intelligence Review* 14, 485–502 (2000)
4. Furnas, G.: Framing the wireless market. *The Future of Wireless*, WSA News:Bytes 17(11), 4–6 (2003)
5. Hadden, J., Tiwari, A., Roy, R., Ruta, D.: Churn prediction using complaints data: Proc. of world academy of science. *Engineering, and Technology* 13, 158–163 (2006)

6. Hadden, J., Tiwari, A., Roy, R., Ruta, D.: Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research* 32, 2902–2917 (2007)
7. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning. Data Mining Inference and Prediction*. Springer, Heidelberg (2008)
8. Hidalgo, C., Rodriguez-Sickert, C.: Persistence, Topology and Sociodemographics of a Mobile Phone Network. Technical report, Center for Complex Network Research, Department of Physics, University of Notre Dame (2007)
9. Katz, L.: A new status index derived from sociometric analysis. *Psychometrika* 18, 39–43 (1953)
10. Kazienko, P., Bródka, P., Ruta, D.: The Influence of Customer Churn and Acquisition on Value Dynamics of Social Neighbourhoods. In: Lytras, M.D., Damiani, E., Carroll, J.M., Tennyson, R.D., Avison, D., Naeve, A., Dale, A., Lefrere, P., Tan, F., Sipiør, J., Vossen, G. (eds.) *WSKS 2009. LNCS*, vol. 5736, pp. 491–500. Springer, Heidelberg (2009)
11. Kim, H., Yoon, C.: Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market. *Telecommunications Policy* 28, 751–765 (2004)
12. Knoke, D.: *Social Network Analysis in Quantitative Applications in the Social Sciences*. Sage Publications, Inc., Thousand Oaks (2007)
13. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A Multi-agent Simulation Environment. *Simulation* 81(7), 517–527 (2005)
14. Newman, M.: The Structure and Function of Complex Networks. *SIAM Review* 45, 167–256 (2003)
15. Onnela, J., Saramaki, J., Hyvonen, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, A.: Structure and Tie Strengths in Mobile Communication Networks. *Proc. Natl. Acad. Sci. USA* 104, 7332–7336 (2007)
16. Vega-Redondo, F.: *Complex Social Networks*. Cambridge University Press, Cambridge (2007)
17. R Development Core Team: *R: A Language and Environment for Statistical Computing*. Foundation for Statistical Computing, Vienna, Austria (2010)
18. Richter, Y., Yom-Tov, E., Slonim, N.: Predicting customer churn in mobile networks through analysis of social groups. In: *SIAM Int. Conf. on Data Mining* (2010)
19. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge (1994)
20. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. *Nature* 393(6684), 409–410 (1998)
21. Yan, L., Miller, D.J., Mozer, M.C., Wolniewicz, R.: Improving prediction of customer behaviour in non-stationary environments. In: *Proc. of International Joint Conference on Neural Networks, IJCNN*, vol. 3, pp. 2258–2263 (2001)

An Agent-Based Platform for Ad-Hoc Social Networking

Vedran Podobnik and Ignac Lovrek

University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
{vedran.podobnik, ignac.lovrek}@fer.hr

Abstract. Social network services such as Facebook, MySpace or Twitter support more or less permanent social relationships, where user interaction takes place via fixed or mobile access to the Internet. However, in many situations social relationships are ad-hoc (i.e., set up by (mobile) users located in a limited geographical area during a certain period in time). To effectively support this kind of social relationships, we propose an extension of basic social networking model – the ad-hoc social networking. This paper defines the model for ad-hoc social networking, describes functionalities and architecture of a platform for ad-hoc social networking and explains why software agents are opportune technology for platform implementation. Finally, paper presents several applications of ad-hoc social networking.

Keywords: ad-hoc connections, social network service, mobile user, user agent, user collaboration, smart community.

1 Introduction

Social Network Service (SNS) on the Internet, supported by web-based platforms such as *Facebook*, *MySpace*, *Twitter* and others, includes display of a user and his/her social connections, as well as additional services that allow him/her to interact through various forms of network-enabled communication. Social relationships established by SNSs are based on ego social networks¹ – every user is building his/her own social network by defining connections with other people. Consequently, a fundamental entity of a social network is an individual. However, SNSs are not focused on the individuals, but on relationships among individuals – they enable information exchange and cooperation among users within a social network which was built by an individual (i.e., within one ego social network), or, among social networks of different users (i.e., among multiple ego social networks). Consequently, user cooperation possesses characteristics of a *smart community* [21], although the initial purpose of social networking was creation of a user’s ego social network. Moreover, if users are

¹ Note that the structure we refer to as an *ego social network* differs from a structure known in literature as an *ego network* [4]. While the ego social network consists from the set of nodes with direct ties to a focal node (called “ego”) and only the set of corresponding ties, the ego network additionally contains the set of ties among non-ego members.

represented within a SNS with their personal software agents, then a multi-agent system implementing a SNS possesses characteristics of a *collective computational intelligence* [20].

SNSs on the Internet support more or less permanent social relationships, where user interaction takes place via fixed or mobile access to the Internet. However, in many situations social relationships are ad-hoc (i.e., set up by (mobile) users located in a limited geographical area during a certain period in time). Such an appearance of a mobile user in a specified location during a specified period in time is often associated with a certain social event (e.g., business/academic event such as a meeting/conference, everyday event such as a football match, or disaster event such as a road accident). To effectively support this kind of social relationships, we propose an extension of basic SNS model – the *ad-hoc social network service* (ahSNS). A platform for ad-hoc social networking should enable three features:

- creation and membership management for an ad-hoc social network (ahSN),
- transfer of user profiles and other user-related information between a SNS and an ahSN, and
- interaction in the ahSN via mobile network.

This paper, i) defines the ahSN paradigm, and, ii) explains why software agents are opportune technology for ahSN implementation. Firstly, Section 2 gives an overview of a related work. Afterwards, Section 3 derives a model of ad-hoc social relationships and Section 4 presents interactions in ahSN. Finally, Section 5 describes the architecture of a platform for ad-hoc social networking, while Section 6 concludes the paper and announces our future work.

2 Related Work

Various combinations of a user:

- location in a particular period of time,
- knowledge/skill/experience/interests, and
- social relationships

are already recognized by SNSs on the Internet as one of the major pillars for their future business models, as well as by academia as a hot research topic.

Facebook, a leading SNS with over 500 million users, introduced *Facebook Places* [8], a supplementary service which enables *Facebook* users to:

- “easily share where they are, what are they doing and the friends they are with right from their mobile”,
- “never miss another chance to connect when they happen to be in the same place at the same time as one of their friends”, and
- “find local deals by checking in to get individual discounts, sharing savings with friends or earning rewards for repeat visits”.

Furthermore, *Foursquare* [9] and *Gowalla* [10], a SNSs devoted exclusively to mobile users, have been in public use since 2009 and already have reached 5 and 1 million users, respectively. During registration procedure a new *Foursquare/Gowalla*

user has an option of importing his/her friends from other SNSs (such as *Facebook* and *Twitter*). *Foursquare/Gowalla* users publish their current location (automatic “checking-in” based on current GPS-location) and post location-related comments. They are encouraged to do that because not only they can easily connect with friends nearby but also they get rewards in form of a certain “badge” (e.g., *Foursquare* user will be the *Major* of a place if most times reported in this area).

There is also a number of related research projects. For example, in [18] authors present a graph analysis based approach to study social networks with geographic information and new metrics to characterize how geographic distance affects social structure. Furthermore, the Stanford’s Mobile and Social Computing Research Group developed Junction [13], an infrastructure designed to support *partyware* (class of social software that assists users in their real-world social encounters). In [11] authors propose the *MobiSoc* middleware which provides a common platform for capturing social state of physical communities by learning patterns from the geo-social data and incorporating this new knowledge with the social state. Another project, called *Social Serendipity* [7], applies Bluetooth technology for detecting other nearby users (by calculating similarity score between user profiles and behavioural data) and uses it to prompt informal interactions between nearby users who do not know each other.

3 A Model of Ad-Hoc Social Relationships

An ahSNS can be created as a *closed* or an *open* SN. A *closed ahSN* includes following two features:

- definition of an ahSN initiator (e.g., user u_i), and
- automatic invitation, by the initiating user u_i , of certain SN-members for joining the $ahSN_i$, their registration to the $ahSN_i$ and, finally, direct interconnection of all users in the $ahSN_i$.

An *open ahSN* includes one additional feature:

- registration of non SN-members to the $ahSN_i$ and direct interconnection with all users in the ahSN. A registration of non SN-members must be approved by the initiating user u_i .

It can be noted that the initiating user u_i manages the $ahSN_i$ ’s membership, regardless of the fact whether the $ahSN_i$ is a closed or an open ahSN.

A graph describing ahSN is a *complete graph*², while a geodetic distance between all pairs of users in the $ahSN_i$ is equal to one (1) – both facts are consequences of direct interconnection of all users in the ahSN. This is also the reason why adjacency matrix [12] of every ahSN is identical – with all non-diagonal elements equal to one (1) and all diagonal elements equal to zero (0). All that simplifies a formal definition of an ahSN – we can define ad-hoc social network $ahSN_i$ as:

$$ahSN_i = (u_i, u_j, u_k, u_l, u_x, u_y). \quad (1)$$

² Complete graph is a simple graph in which every pair of distinct users is connected.

In the $ahSN_i$ defined with equation (1) user u_i is an initiating social network member, while users u_j, u_k, u_l, u_x and u_y are non-initiating social network members.

A user is represented within an ahSN with a profile, according to the following rule:

- *SN-members of an ahSN* – ahSN-profile is imported from a user's SN-profile (under the control of the user), and
- *non SN-members of an ahSN* – user creates an ahSN-profile from the scratch.

A member of an ahSN should be able to export his/her ahSN-profile to a certain permanent SN, such as *Facebook*.

4 Interactions in an Ad-Hoc Social Network

Various SNSs support different modes of interactions. For example, *Facebook* enables its users very rich selection of communicating mechanisms – they can interact both asynchronously (via *private messages* to another user/group of users by using *Messages* application; or via *public messages* to another user by writing to that user's *Wall* application) and synchronously (via *private messages* to another user by using *Chat* application). On the other hand, *Twitter* enables its users only to post and read short messages (up to a maximum of 140 characters long) called *tweets*.

An ahSN is an ad-hoc community in which interactions can be twofold:

- *each-to-each communication* – public message published by one ahSN-member and broadcasted to all other members of the ahSN, and
- *one-to-one communication* – private message sent by one ahSN-member to another.

The basic form of interactions is via public messages, because an ahSN is established for certain purpose or reason which is common to all members. Public messages are published via a common available medium and broadcasted to all ahSN-members (alternatively, ahSN-members can only be informed about the publication of messages via a message notification). Message notifications, as well as public and private messages, could be distributed in the mobile network via SMS (*Short Message Service*) or MMS (*Multimedia Messaging Service*).

User interaction within an ahSN should have i) *better availability*, ii) *lower costs*, and iii) *greater energy efficiency* than those offered by “ordinary” SNSs. For all three reasons, it is rational to locate the ahSN server closer to users and make it almost immediately available for the mobile network that geographically covers the location relevant for the ahSN.

Service availability is critical for situations such as (natural) disasters, when the global connectivity required for SNSs, as well as communication in general, is difficult or impossible to achieve. Consequently, ahSNSs can serve as support for the disaster management processes.

Service costs are reduced because users need to connect to the SNS web-site only while creating an ahSN (i.e., during invitation and registration of new ahSN-members, including the import of users' SN-profiles).

Through a reduction of the number of communication and server systems mediating the service, achieved through replacing a remote service provisioning paradigm

with a local service provisioning, ahSNSs become *green services* characterized with lower energy consumption than “ordinary” SNSs. Additionally, energy efficiency of ahSNSs can be additionally increased by using short-range communication technologies (e.g., Bluetooth) for communication among ahSN-members.

Furthermore, user interaction within an ahSN can be enhanced by supplementary services, such as:

- *meeting and group facilitation* – scheduling of activities, temporal and spatial arrangement of the group according to specific knowledge/skill/experience,
- *collaborative download* – piecewise download of a specific document by group members and mutual exchange of pieces in order to merge them into a full document,
- *social search* – search for new members fulfilling some knowledge/skill/experience criteria or sharing some common interest, and
- *buyer coalition formation* – grouping members in order to reduce price or take advantage of volume discounts.

5 A Platform for Ad-Hoc Social Networking

We have defined a model of ad-hoc social relationships, procedures for building an ahSN and rules for communication among ahSN-members. Now we will describe architecture of a platform for ad-hoc social networking and present several ahSN services we designed and implemented.

5.1 An Agent-Based Architecture of a Platform for Ad-Hoc Social Networking

From the architectural perspective, a platform for ad-hoc social networking consists of following components (Fig. 1):

- Client application – *ahSNS application*,
- Server with basic and supplementary services – *ahSNS server*,
- Social network database – *ahSNS application database*, and
- Client application distribution entity – *ahSNS application distribution store*.

Users interested in using ahSNSs firstly have to download the *ahSNS application* from the *ahSNS application distribution store* (e.g., *iTunes Store*, *Android Market*, etc.). Now they can invite other users (or be invited) to build a certain ahSN, according to procedures defined in Section 3. While registering to the ahSN users SN-members can import their profiles from existing SNSs (e.g., *Facebook*, *Twitter*, etc.). When registered to certain ahSN, users can utilize its basic and supplementary services via mobile network.

Software agents are opportune technology for the platform implementation because they enable high-level of automation for users while preserving their privacy (regardless mobile device they use), as well as seamless communication through the network (they take care of all protocol adjustments, e.g., SIP-to-ACL (*Session Initiation Protocol* to *Agent Communication Language*) message conversion) [1]. The following types of agents are needed for the platform implementation:

- *User Agents (UA)* – implemented as a part of ahSNS application. User Agents enable their owners (semi-)automated interaction with ahSNS Agent. Moreover, as they are trusted entities for their owners, they enable automated profile import/export between SNSs (for users SN-members) and the ahSNS. User Agents can learn user preferences and habits over time and adapt according to them if necessary.
- *ahSNS Agent (SA)* – implemented as a manager of ahSNS server. The ahSNS Agent coordinates activities within the ahSN server, gives tasks to specialized agents and serves as a gateway between User Agents and the ahSN server, and
- *specialized agents* – implemented as a part of ahSNS server, every specialized agent is in charge of one basic or supplementary service. Specialized agents receive tasks from the ahSNS Agent, execute them, refresh the ahSN database and return the result to the ahSNS Agent (if needed).

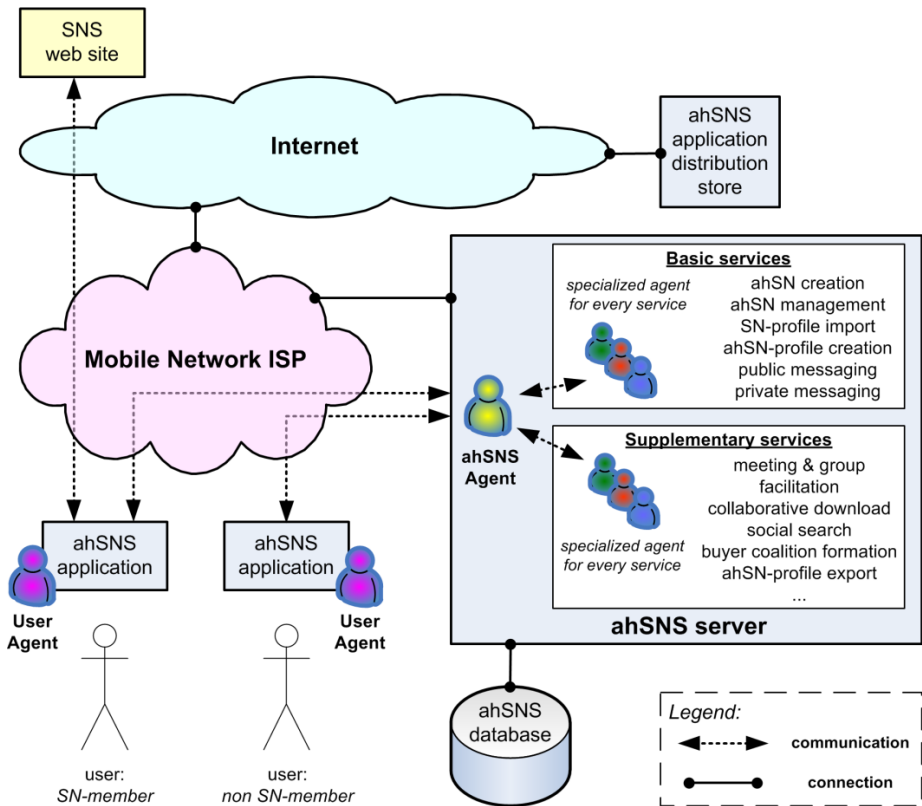


Fig. 1. A platform for ad-hoc social networking

5.2 Proof-of-Concept Ad-Hoc Social Network Services

We already used JADE agent platform to implement a number of ahSN services. We will present multi-agent systems that enable: i) session mobility in next-generation

network (NGN), ii) group-oriented service provisioning in NGN, and, iii) *Collaborative Urban Computing* (CUC) service.

Agent-based Session Mobility in Next-Generation Network. Session (mid-call) mobility occurs when a user changes a terminal, moves to another network, or switches to another network interface during an on-going session. After the mobile user has obtained a new IP (Internet Protocol) address it re-invites the correspondent host in order to re-establish communication.

Possible ahSNS scenario can be the following. Agent UA_{Bob} , Bob's personal agent, i) enables specialized agent on ahSNS server (e.g., Viber Agent, VA) to track Bob's location, and, ii) activates a rule according to which VA creates and manages ahSN called "Bob's Viber Network". This ahSN consists of all Bob's contacts from his iPhone phonebook who have currently active Viber³ application. At the same time Bob, who also installed Viber application on his iPhone, joins ahSN "Bob's Viber Network" whenever he is at work or at home (i.e., places with WiFi coverage). The VA ensures that Bob, when at work or at home, conducts all his calls to other users from "Bob's Viber Network" via Viber application, consequently reducing Bob's calling costs. Moreover, VA is also capable of autonomously converting non-Viber calls into Viber-calls (and vice versa), as a consequence of i) Bob's mobility, or, ii) change in the structure of ahSN "Bob's Viber Network".

We utilized the LocalNote service [5], a location-triggered instant messaging service that provides a mechanism for sending text messages whereby the sender can specify the area in which the recipient must reside in order to receive the message, to enable described agent-based session mobility in NGN [15].

Agent-enabled Group-Oriented Service Provisioning in Next-Generation Network. Superdistribution [19] is the combined distribution and market scheme for digital goods involving buyers (i.e., end-users) in the distribution process in such a way that they redistribute a good to other legitimate buyers. The principle of superdistribution is tightly connected with the viral marketing phenomenon [14]. Superdistribution can be explicit (end-users exchange digital goods) or implicit (end-users exchange recommendations for digital goods, while digital goods are distributed by businesses that act as content providers (e.g., via *iTunes Store*)).

We implemented a JADE multi-agent system for group-oriented service provisioning in NGN [16][17]. Specialized agents were in charge of: i) user profile management, ii) user profile semantic comparison, iii) SN creation and analysis (i.e., user clustering/classification). As a proof-of-concept we implemented a group-oriented service RESPIRIS (*Recommendation-based Superdistribution of Digital Goods within Implicit Social Networks*). The innovativeness of the RESPIRIS service lies within the fact that it represents the implicit superdistribution scheme - superdistribution groups (i.e., ahSNS) are generated dynamically and implicitly based on mobile user profiles. Namely, each subscriber of the proposed service is represented by a corresponding profile wherein his/her preferences are described (e.g., what sort of digital good he/she is interested in, of which genre, etc.). User profiles are used for semantic clustering of

³ Viber is an iPhone application that lets users make free phone calls to other iPhone users that have Viber installed (<http://www.viber.com>).

users into superdistribution groups. The groups users are allocated into are based on users' preferences' similarities and built autonomously.

Possible ahSNS scenario can be the following. If a particular user, e.g., Bob, is interested in the RESPIRIS service then his personal agent UA_{Bob} sends a registration request to the specialized agent on ahSNS server (e.g., RESPIRIS Agent, RA). The RA contains a list of m user profiles whose users subscribed to the RESPIRIS service and allocates these profiles into k groups according to users' preferences – every group representing one ahSN. In a particular ahSN _{x} consisting of n ($n < m$) users, the RA informs each of the n UAs of all other UAs within that particular ahSN. Now, when Bob purchases some digital content, all other users from the same ahSN are notified. Moreover, UA_{Bob} can provide other UAs from the same ahSN with information about Bob's usage of a purchased digital content (e.g., a frequency of reproducing a newly purchased song) or his level of satisfaction with it (e.g., Bob's explicit rating of a newly purchased song).

Described approach in digital goods distribution ensures that all involved parties benefit from interactions between ad-hoc groups of end-users. Namely, let alone the fact that providing digital goods does not imply almost any production expenses, the content provider need not perform resource-consuming tasks such as advertisement – the advertisement becomes redundant as users receive recommendations from other users in the same ahSN. On the other hand, end-users benefit from this solution as it facilitates choosing from a large number of generally available digital goods, as they are provided with the ability to concentrate on a smaller number of recommended goods. What makes these goods potentially interesting is the fact that recommendations are sent by users from the same ahSN, which is assembled according to the similarity in preferences of users whose profiles it encompasses.

Agent-enabled Collaborative Urban Computing. The main idea behind Collaborative Urban Computing (CUC) [3] is that a set of users, physically located in an urban environment, can serendipitously cooperate using their computing devices (e.g., mobile phones or laptops) to achieve a common goal. The idea of CUC is related to the opportunistic computing paradigm, where different types of devices with networking capabilities represent distributed resources which can be pooled together and used collectively to improve the overall user Quality of Experience (QoE) [6]. We implemented a multi-agent system enabling a proof-of-concept CUC service, which we named the *Collaborative Downloading*. The Collaborative Downloading service replaces today's standard approach to mobile service provisioning (i.e., the individual approach) with the novel collaborative approach. In the individual approach, mobile users communicate only with the content server and download the entire desired content autonomously using certain mobile network technology. On the other hand, the collaborative approach allows mobile users interested in the same content to collaborate and download the desired content together. Content on the content server is divided into several parts and each part can be downloaded independently from the server via certain mobile network technology (e.g., *General Packet Radio Service*, GPRS). After downloading, these parts can be exchanged with other mobile users using an ad-hoc network technology (e.g., Bluetooth).

Possible ahSNS scenario can be the following. If a particular user, e.g., Bob, is interested in the Collaborative Downloading service then his personal agent UA_{Bob}

sends a registration request to the specialized agent on the ahSNS server (e.g., Collaborative Download Agent, CDA). Together with the registration, UA_{Bob} authorizes CDA to track Bob's location. Consequently, the CDA contains a list of m user profiles whose users subscribed to the Collaborative Downloading service, as well as their current locations. The whole system is on hold until one of the registered users (i.e., Bob) requires a certain downloading service. Now, the CDA firstly filters all registered users according to Bob's current location and then identifies a group of n ($n < m$) users representing Bob's "ahSN for collaborative downloading", who are invited to collaborate (according to rules defined in [2]) while downloading the content Bob is interested in. At the end, computing devices of all users from Bob's "ahSN for collaborative downloading" who agreed to collaborate contain the entire content Bob was interested in.

6 Conclusions and Future Work

Recent technical advancements in computing devices and communication networks changed the way people use Information and Communication Technology (ICT) systems. One of the most obvious examples is rapid proliferation of the social networking services (SNSs) during the last decade. As a result, today there exist a myriad of SNSs, which all support more or less permanent social relationships. However, SNSs will evolve from ego-centric social networking systems into smart community systems by taking into account spatial and temporal information about users, as well as having in mind their specific knowledge/skill/experience/interests.

In this paper, we presented the ad-hoc social network (ahSN) paradigm which supports spontaneous cooperation between a set of users physically located in a limited geographical area during a certain period in time and all sharing a common goal. Furthermore, we described functionalities and architecture of a platform for ad-hoc social networking and gave a brief overview of platform's possible applications.

For future work, we plan to implement a platform for ad-hoc social networking using software agents and extend a list of platform's supplementary services.

Acknowledgements. The authors acknowledge the support of research project "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Belavic, R., Basuga, M., Podobnik, V., Petric, A., Lovrek, I.: Agent-based Social Networking for Mobile Users. *International Journal of Intelligent Information and Database Systems* 4(6), 599–628 (2010)
2. Bojic, I., Podobnik, V., Kusek, M.: Agent-enabled Collaborative Downloading: Towards Energy-efficient Provisioning of Group-oriented Services. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6071, pp. 62–71. Springer, Heidelberg (2010)
3. Bojic, I., Podobnik, V., Kusek, M., Jezic, G.: Collaborative Urban Computing: Serendipitous Cooperation between Users in an Urban Environment. *Cybernetics and Systems* (in press)

4. Borgatti, S.P., Mehra, A., Brass, D., Labianca, G.: Network Analysis in the Social Sciences. *Science* 323(5916), 892–895 (2009)
5. Brajdic, A., Lapevic, O., Matijasevic, M., Mosmondor, M.: Service composition in IMS: A location based service example. In: Proc. of the 3rd Int. Symposium on Wireless Pervasive Computing (ISWPC 2008), Santorini, Greece, pp. 208–212 (2008)
6. Conti, M., Giordano, S., May, M., Passarella, A.: From Opportunistic Networks to Opportunistic Computing. *IEEE Communications Magazine* 48(9), 126–139 (2010)
7. Eagle, N., Pentland, A.: Social Serendipity: mobilizing social software. *IEEE Pervasive Computing* 4(2), 28–34 (2005)
8. Facebook places, <http://www.facebook.com/places> (visited on: 02.01.2011)
9. Foursquare, <http://foursquare.com> (visited on: 02.01.2011)
10. Gowalla, <http://gowalla.com> (visited on: 02.01.2011)
11. Gupta, A., Kalra, A., Boston, D., Borcea, C.: MobiSoC: a middleware for mobile social computing applications. *Mobile Networks and Applications* 14(1), 35–52 (2009)
12. Jackson, M.O.: *Social and Economic Networks*. Princeton University Press, Princeton (2008)
13. Junction, <http://mobisocial.stanford.edu/index.php#junction> (visited on: 02.01.2011)
14. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. *ACM Transactions on the Web* 1(1), art. no. 5 (2007)
15. Petric, A., Trzec, K., Jurasovic, K., Podobnik, V., Jezic, G., Kusek, M., Ljubi, I.: Agent-based Support for Context-aware Provisioning of IMS-enabled Ubiquitous Services. In: Kowalczyk, R., Vo, Q.B., Maamar, Z., Huhns, M. (eds.) *SOCASE 2009*. LNCS, vol. 5907, pp. 71–82. Springer, Heidelberg (2009)
16. Podobnik, V., Galetic, V., Trzec, K., Jezic, G.: Group-Oriented Service Provisioning in Next Generation Network. In: Srinivasan, D., Jain, L.C. (eds.) *Innovations in Multi-Agent Systems and Applications*, Studies in Computational Intelligence, vol. 310, pp. 277–298. Springer, Heidelberg (2010)
17. Podobnik, V., Petric, A., Trzec, K., Galetic, V., Jezic, G.: Agent-Based Provisioning of Group-Oriented Non-linear Telecommunication Services. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009*. LNCS, vol. 5796, pp. 193–204. Springer, Heidelberg (2009)
18. Scellato, S., Mascolo, C., Musolesi, M., Latora, V.: Distance Matters: Geo-social Metrics for Online Social Networks. In: Proc. of the 3rd Conf. on Online Social Networks (WOSN 2010), Boston, USA (2010)
19. Schmidt, A.U.: On the Superdistribution of Digital Goods. *Journal of Universal Computer Science* 15(2), 401–425 (2009)
20. Singh, V.K., Gautam, D., Singh, R.R., Gupta, A.K.: Agent-based Computational Modeling of Emergent Collective Intelligence. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009*. LNCS, vol. 5796, pp. 240–251. Springer, Heidelberg (2009)
21. Venezia, C., Taylor, N., Williams, H., Doolin, K., Roussaki, I.: Novel Pervasive Computing Services Experienced through Personal Smart Spaces. In: Proc. of the Int. Workshop on Data Management in Ad Hoc and Pervasive Computing in conjunction with the 10th Int. Conf. on Mobile Data Management (MDM 2009), Taipei, Taiwan, pp. 484–489 (2009)

Formal Specification of Multi Agent System for Historical Information Retrieval from Online Social Networks

Ruqayya Abdulrahman, D.R.W. Holton, Daniel Neagu, and Mick Ridley

Department of Computing, University of Bradford, Bradford, U.K.
{rshabdul, d.r.w.holton, d.neagu, m.j.ridley}@bradford.ac.uk

Abstract. In the last few years, there has seen a marked increase in the usage of Online Social Networks (OSN). Coping with the real time changes of a huge amount of personal information requires new methods for retrieving information. Our previous approach for retrieving information from OSN was a centralized system that controlled and retrieved information of each user's profile once. In order to enhance that approach, a Multi Agent System (MAS) was chosen for its ability to deploy agents to monitor profiles continuously. Since MAS is difficult to implement correctly, a formal specification of an OSN System was constructed to help us investigate the feasibility of using MAS technology in retrieving historical information from OSN sites.

Keywords: formal specification, agent, multi agent system, information retrieval, online social network.

1 Introduction

It is becoming increasingly difficult to ignore the huge amount of continuously changing personal information in Online Social Networks (OSNs) such as Myspace, Facebook and Twitter. Since our research aims to retrieve information from OSN, new methods are required to keep pace with the real time changes in OSN profiles. In our previous approach [1], we presented an algorithm for automated extraction of information from an OSN. However, that approach has several sources of limitations which need to be addressed. The main cause of those limits lies in the fact that the system was designed as a centralized system. As a result, the system failed if it was not possible to extract information from a profile if the profile's structure did not match the cases in our parser. Moreover, the previous algorithm was time consuming and did not address monitoring profiles over time.

To overcome these barriers, a Multi Agent System (MAS) approach was selected because of its characteristics such as mobility, autonomy, sociability and perceptivity [2,3]. MAS offers the ability to retrieve information from an OSN in a parallel (multithread) approach. However, the fact that MAS is considered to be one of the most complex systems to develop [4,5] adds extra pressure on software developers when they decide to use one. They must avoid issues such as running over time,

exceeding budget, producing incorrect or inefficient software products and not being able to reuse the software components. Mostly, these problems are caused as a result of choosing informal, imprecise methods at different stages of the software engineering process. Thus, using natural language alone to specify the purpose of MAS in designing Online Social Network Retrieval System (OSNRS) is not sufficient. Natural language is vague, inaccurate and can tolerate more than one interpretations [6]. Usually, this is the main cause of producing ambiguous and incomplete systems' specifications.

This paper proposes the use of MAS technology in retrieving information from OSN sites. The proposal is explored by means of constructing a formal specification of a MAS framework. The structure of the paper is as follows: section 2 presents a brief background of formal specification notations. Section 3 highlights a detail description of the formal specification of our proposed system (OSNRS). Section 4 presents the conclusion and future works.

2 Background and Related Work

2.1 Formal Specification Notations

Several definitions have been stated to define formal specification. They differ according to the domain. In [7], Baryannis presents some of them. The definition that fits with our OSNRS is the one that is in the context of Web services which identifies the formal specification as “*a precise mathematical description, supported by a sound logic and reasoning apparatus, of what the service should do in terms of its capabilities and the functionality it offers*”. Various research approaches have been developed to define new formal languages to specify Agents and MAS. e.g. Logic for Contract Representation (LCR) language [8] is developed to describe interaction in MAS using a very expressive logic, SLAB which stands for Specification Language for Agent-Based Systems [9,4,10] focused on how agents' behaviours relate to the environment. SLAB produced the concept of Caste i.e. a set of agents in MAS that have common capability and behaviour to perform a certain task.

When Niazi and Hussain in [11] developed a simulation model for wireless sensor networks for monitoring Complex Adaptive Environment, they chose the formal specification language Z to present a novel Formal framework for Agent-Based Simulation (FABS). Z has been made as ISO standard. Thus it is used widely in academia as well as in large projects in industry [12-14].

This paper follows the approach of [5] and [15] in which Object Z was used in formalizing MAS. Object Z is an extension of Z with supporting object-oriented specification. Object Z has a class construct that combines the state schema and its relevant operations. It supports all MAS features covered by our proposed system (OSNRS) specifically concurrency, communications and state. Also multiple communication could be applied through range of composition operators [16]. This fits well with our proposed system (OSNRS) that will use Java to implement MAS.

3 Formal Specification of OSNRS

Before giving the details of the formal specification of our proposed system using Object Z, we will present a simple scenario to illustrate the behaviour of OSNRS. Then we will illustrate the basic types of the systems followed by the schemas described in detail. The terms *user profile*, *URL address* and *identity* are used interchangeably in this paper.

3.1 Problem Scenario

As illustrated in Fig.1, OSNRS starts when the user specifies an identity profile to retrieve its information. The MasterAgent (*mAg*) takes this identity and assigns it to a grabber agent (*gAg*). The *gAg* will retrieve information such as name, gender, age, country, etc. as well as a list of the profile’s top friends. This information will be stored locally to be compared with the information retrieved the next time the agent is activated. Also the information will be sent to the *mAg* and used to create a history of each profile in a local repository for future analysis.

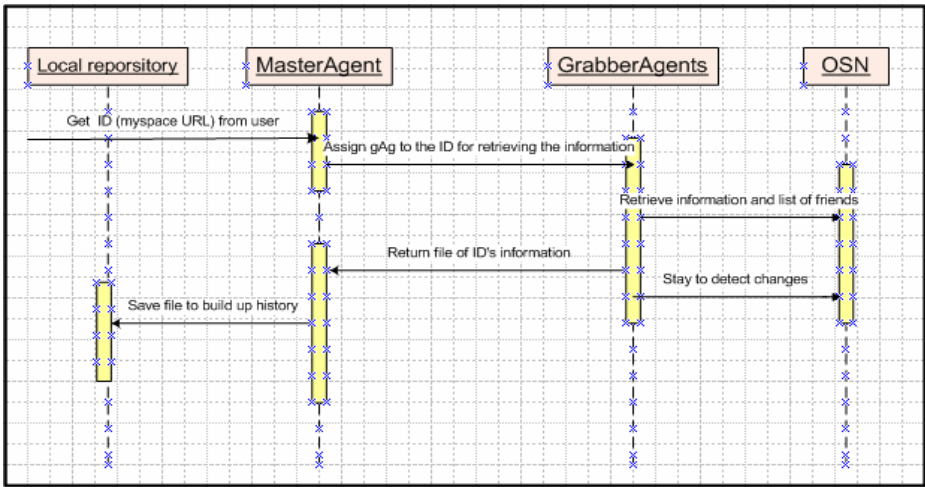


Fig. 1. Sequence diagram of OSNRS

The *mAg* will assign a *gAg* to each friend in the new top friends list, provided it has not already been assigned a *gAg*. The retrieval cycle will repeat until the *mAg* matches one of the stopping criteria as will be described in section 3.3.1. Each *gAg* will be activated periodically to detect updates in the profile that it is assigned to.

3.2 OSNRS Basic Types

From the OSNRS described scenario, we identify the basic types of our system as: $[Agent, Container, Record, Identity]$ where $[Agent]$ the set of agents.

[*Record*] the set of files that contains the retrieved information from the users' profiles excluding the list of top friends.

[*Identity*] the set of the identities of OSN profiles.

3.3 The Objects of OSNRS

The most important components of OSNRS are its two main objects: MasterAgent and GrabberAgent. They are detailed separately in the sections below:

3.3.1 MasterAgent Object

A. Global Constants:

$iterationLevel: \mathbb{N}$	$counter: \mathbb{N}$
------------------------------	-----------------------

The object MasterAgent has two constant variables. Both are specified by the user to declare the stopping criteria: the *iterationLevel*: is a natural number to declare how depth the extraction process will be. E.g. friends of friends.

The second constant is *counter*: to declare the number of friends to extract their information regardless if they are a friend or a friend of a friend.

B. State schema:

$known : iseq_1 Identity$ $waitingResponse: P Identity$ $visiting : Agent \rightsquigarrow Identity$ $current, next : (iseq Identity \times \mathbb{Z})$
$waitingResponse \subseteq ran known$ $ran first next \subseteq ran known$ $ran first current \subseteq ran known$ $disjoint \langle ran first current, ran first next \rangle$

The state schema in the class MasterAgent contains the state variables as illustrated below:

- *known*: is an injective sequence of all identities (URLs) that are known to the MasterAgent, either the identity has been retrieved as a friend in top list or entered by the user. Each identity in *known* is unique and at least has one identity which is the *seed*? that is entered by the user.
- *waitingResponse*: is a set of identities that grabber agents have been assigned to, but whose details have not yet been retrieved.
- *visiting*: is a one to one relationship between Agent and Identity to record which identity a grabber agent is visiting. Because it is a 1:1 relationship, each profile identity is visited just by at most one grabber agent.
- *current and next*: are sequences. *Current* is used to determine the identities that are being retrieved in this iteration level while *next* represents the identities that will be retrieved in the next iteration level. i.e when the extraction process moves from an iteration level to another level, the contents

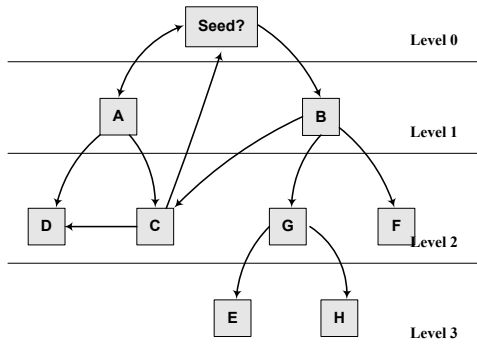
of *next* will become as *current* and *next* will contain the list of top friends of identities in the *current*.

The state invariants points out two roles of *known*; first of all, it has to record, in addition to the *seed?*, all retrieved identities. The second role is to filter the retrieved identities before adding them to the existing list. Thus, all identities in *next*, *current* and *waitingResponse* is taken from *known*. Since in any level, the identities that are in *next* will become in the *current* for the next levels, this will ensure that the identity will not be extracted twice by different grabber agents as will be illustrated in Fig. 2.

Informal Description Example

To simplify understanding the scenario of OSNRS and its schemas, let’s consider the following graph and table (see Fig. 2). The *seed?* node is the identity (URL address of the profile) that is entered by the user. It will be added to the *known* as shown in level 0 in the table. This node represents the root and it will be in *next* to determine that the tuple $\langle\langle seed? \rangle, 0\rangle$ will become the *current* tuple for extraction in the level 1.

The *seed?* node has two top friends in its list which are A and B. These two nodes will be added to the *known* and compose the tuple $\langle\langle A, B \rangle, 1\rangle$ to be extracted in level 2. Node A has three friends in its top friends list, which are C, D and *seed?*. Therefore, the link between *seed?* and A is bidirectional, i.e. they are in the top friends list of each other. However, *known* will filter the retrieved identities to not allow any redundancy, i.e. *seed?* will not be added. The same principle is applied to the subsequent levels.



Iteration Level	<i>current</i> '	<i>known</i> '	<i>next</i> '
Initial	\emptyset	$\langle \rangle$	\emptyset
0	\emptyset	$\langle seed? \rangle$	$\langle\langle seed? \rangle, 0\rangle$
1	$\langle\langle seed? \rangle, 0\rangle$	$\langle seed? \rangle \overset{\curvearrowright}{\langle A, B \rangle}$	$\langle\langle A, B \rangle, 1\rangle$
2	$\langle\langle A, B \rangle, 1\rangle$	$\langle seed?, A, B \rangle \overset{\curvearrowright}{\langle D, C, G, F \rangle}$	$\langle\langle D, C, G, F \rangle, 2\rangle$
3	$\langle\langle D, C, G, F \rangle, 2\rangle$	$\langle seed?, A, B, D, C, G, F \rangle \overset{\curvearrowright}{\langle E, H \rangle}$	$\langle\langle E, H \rangle, 3\rangle$

Fig. 2. A sample model of Online Social Network and the changes in the schema’s attributes

C. The initial schema

<p style="text-align: center;">INIT</p> <p>$known = \langle \rangle$ $waitingResponse = \emptyset$ $visiting = \emptyset$ $current = \emptyset$ $next = \emptyset$</p>
--

When the class is initialized, the known, current and next are empty sequences and waitingResponse and visiting are empty sets.

MasterAgent Operations:

There are seven main operations that make up the MasterAgent as presented below:

D. The start operation

$$start \hat{=} Init \text{ } \textcircled{\&} \text{ } [seed? \mid known = \langle seed? \rangle \wedge \\ waitingResponse = \{ seed? \} \wedge \\ next = (\langle seed? \rangle, 0)]$$

When the system is started for first time and the user entered the identity $seed?$, the mAg has to add this $seed?$ to the sequence $known$ that holds all identities that the system will retrieve its information. Also, $seed?$ will be added to $waitingResponse$ set to indicate that this identity is assigned to a gAg but has not retrieved its information yet. Consequently, the variable $next$ will contain the tuple $(\langle seed? \rangle, 0)$ to specify that the $seed?$ will become the next step for extraction.

E. The assignGrabber operation

As described in the scenario, the first operation the MasterAgent object has to do when it receives the $seed?$ from user is to assign it to a grabber agent. The $assignGrabber$ is a recursive operation that composes itself with $addAssigning$ operation. It will continue until all identities have been assigned.

$$assignGrabber \hat{=} addAssigning \text{ } \textcircled{\&} \text{ } assignGrabber \\ [] \\ [first\ next = \langle \rangle] \text{ } receiveAllProfiles$$

F. The addAssigning operation

To accomplish $addAssigning$ operation, we have to check that there exists a grabber agent (gAg) from the class GrabberAgent which has not been assigned before to any identity. Once $addAssigning$ achieved successfully, the visiting after state will be updated to include the $\{gAg \mapsto id!\}$.

<i>addAssigning</i>
Δ <i>visiting</i> , <i>next</i> <i>id!</i> : <i>Identity</i>
$\exists gAg: GrabberAgent \bullet gAg \notin ran\ visiting \Rightarrow$ <i>visiting'</i> = <i>visiting</i> \cup { <i>gAg</i> \mapsto <i>id!</i> } \wedge <i>next'</i> = (<i>tail first next</i> , <i>second next</i>) \wedge <i>id!</i> = <i>head (first next)</i> \wedge <i>gAg</i> . <i>receiveID</i>

The head of the sequence *next* will be the output of this operation schema (*id!*). Note that we are still in the same level of iteration because the *second* of *next* has not changed. This output (*id!*) will be passed to *GrabberAgent* for retrieving information as will be explained in section 3.2.2. The sequence *known* and the set *waitingResponse* will not be affected.

G. The receiveAllProfiles operation

$$receiveAllProfiles \hat{=} receiveProfiles \otimes receiveAllProfiles$$

$$\square$$

$$[waitingResponse = \emptyset] completeRetrieving$$

Similar to *AssignGrabber*, the *receiveAllProfiles* is a recursive operation that composes itself with *receiveProfiles* operation. But, it will not move to the next level of iteration until the information about all identities in *waitingResponse* is returned.

H. The receiveProfiles operation

When the *MasterAgent* receives a profile from a *grabber agent*, it has to pass this information in addition to the *id?* to the *Database* class for updating the history of this profile. Then it has to delete this *id?* from *waitingResponse*

<i>receiveProfiles</i>
Δ <i>known</i> , <i>waitingResponse</i> , <i>next</i> <i>id?</i> : <i>Identity</i> <i>newRec?</i> = <i>Record</i> <i>newFriends?</i> = <i>seq Identity</i>
<i>let filtered</i> : <i>seq Identity</i> \wedge <i>filtered</i> = (<i>newFriends?</i> \uparrow (<i>Identity</i> \setminus <i>ran known</i>)) <i>known'</i> = <i>known</i> \frown <i>filtered</i> <i>next'</i> = <i>newFriends?</i> <i>waitingResponse'</i> = <i>waitingResponse</i> \setminus { <i>id?</i> }

I. The completeRetrieving operation

This operation either applies the *nextLevel* operation recursively or exits and finishes the iteration if the counter of *iterationLevel* matches the number in the *second* of *next* relation.

$$\begin{aligned}
 \text{completeRetrieving} &\hat{=} \text{nextLevel} \text{ } \text{;} \text{ assigneGrabber} \\
 &\quad \square \\
 &\quad [\text{second next} = \text{iterationLevel}] \text{ Init} \\
 &\quad \square \\
 &\quad [\text{counter} \geq \text{size Known}] \text{ Init}
 \end{aligned}$$

J. The nextLevel operation

When the MasterAgent completes retrieving all information about all identities in the *waitingResponse* set, i.e. completes the iteration level, it has to move to the next level of iteration. In this stage, all identities in *next* will be copied to both *waitingResponse* and *current* to start the new level of retrieving information.

<p style="text-align: center;"><i>nextLevel</i></p> <hr/> <p>Δ <i>current</i>, <i>next</i>, <i>waitingResponse</i> <i>id!</i>: seq Identity</p> <hr/> <p>$\text{next}' = (\langle \rangle, \text{second next} + 1) \wedge$ $\text{current}' = \text{next} \wedge$ $\text{waitingResponse}' = \text{ran first next} \wedge$ $mAg . \text{assignGrabber}$</p>
--

In OSNRS, we create a special agent called MasterAgent (*mAg*) which lives in a container called home. *mAg* is unique and should be active all the time.

<p><i>mAg</i>: Agent <i>home</i>: Container</p>
--

3.3.2 GrabberAgent Object

The state schema for this object contains three state variables: the identity of each grabber agent (*myID*), the personal information of each profile (*myRec*) and the list of top friends (*myFriends*). The only constraint of this schema is that each identity is not allowed to be a friend of itself. When the grabber agent is created, all attributes are initialized to empty sets and sequence respectively.

When the MasterAgent assigns an identity to a grabber agent, it will send the identity through *addAssigning* operation to be received by *receiveID* operation in GrabberAgent class. If the assigning is made for the first time, GrabberAgent will call its initial state. Otherwise, it will call *updateID* to replace the old identity with the

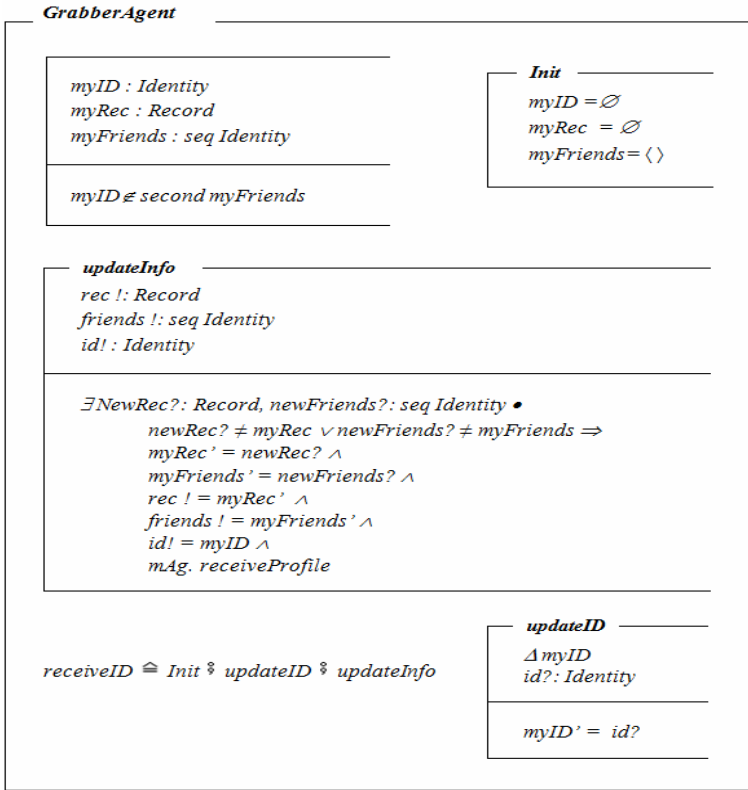


Fig. 3. GrabberAgent Object

new identity. This is important to re-use the grabber agent in cases where the identity that it is assigned to was removed from the sub network of social network. In both situations, the grabber agent has to call the *updateInfo* operation. Since the retrieved information is coming from an external database (MySpace social network), we will assume that there exists a record *newRec?* that contains all personal information about *myID* and a sequence of identities *newFriends?* that holds a list of *myID* top friends. Either the *newRec?* or *newFriends?*, or both should be distinguished from the existing *myRec* and *myFriends* respectively to be replaced. As a result, the new information in *myRec*, *myFriends* accompanied with *myID* will be sent to the MasterAgent object through *rec!*, *friends!* and *ID!* consecutively when *mAg.receiveProfile* operation is used.

4 Conclusions

This paper has constructed a formal specification of an OSNRS to help us to have a better understanding of the feasibility of using MAS technology in retrieving historical information from OSN. It facilitated the description of the architecture of the proposed system by reflecting the required classes to be used in implementation.

Using Object z to capture the system in detail, helped in analyzing the system behaviour. Although Object Z does not explicitly provide a process construct, such as is

present in process algebras like the pi-calculus, the class construct was a suitable replacement. Operation composition allowed message passing between objects to be modelled, and defining a collection of mutually recursive operations modelled the grabbing of profile information intuitively. The OSNRS is implemented according to the constructed formal specification. Full description of the OSNRS is shown in [19].

References

1. Alim, S., AbdulRahman, R., Neagu, D., Ridley, M.: Data retrieval from online social network profiles for social engineering applications. In: The 4th International Conference for Internet Technology and Secured Transactions ICITST, pp. 1–5. IEEE, London (2009)
2. Bordini, R.H., Wooldridge, M.J., Heubner, J.F.: Building multi-agent systems: a practical introduction with Jason. Wiley series in agent technology. John Wiley, Chichester (2007)
3. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. John Wiley & Sons Ltd., Chichester (2007)
4. Zhu, H.: A formal specification language for agent-oriented software engineering. In: the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1174–1175. ACM, New York (2003)
5. Hilaire, V., Koukam, A., Gruer, P., Müller, J.-P.: Formal specification and prototyping of multi-agent systems. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2000. LNCS (LNAI), vol. 1972, pp. 114–127. Springer, Heidelberg (2000)
6. Currie, E.: The essence of Z. Prentice Hall Europe (1999)
7. Aspray, W., Keil-Slawik, R., Parnas, D.: History of Software Engineering (1996), <http://www.dagstuhl.de/Reports/96/9635.pdf>
8. Potter, B., Sinclair, J., Till, D.: An introduction to formal specification and Z. Prentice-Hall, Englewood Cliffs (1991)
9. Baryannis, G.: Formal Specification, definitons (2010), <http://www.s-cube-network.eu/km/terms/f/formal-specification> (accessed 21-12-2010)
10. Dignum, V., Meyer, J., Dignum, F., Weigand, H.: Formal specification of interaction in agent societies. In: Formal Approaches to Agent-Based Systems, pp. 37–52 (2002)
11. Zhu, H.: A formal specification language for MAS engineering. In: The 2nd International Workshop on Agent-Oriented Software Engineering (2001), <http://cms.brookes.ac.uk/staff/HongZhu/AFM/PublicationArchive/AOSE2001CMR.pdf>
12. Zhu, H.: Formal specification of agent behaviour through environment scenarios. In: Rash, J.L., Rouff, C.A., Truszkowski, W., Gordon, D.F., Hinchey, M.G. (eds.) FAABS 2000. LNCS (LNAI), vol. 1871, pp. 263–277. Springer, Heidelberg (2001)
13. Niazi, M., Hussain, A.: A Novel Agent-Based Simulation Framework for Sensing in Complex Adaptive Environments. IEEE Sensors Journal, 404–412 (2010)
14. Jacky, J.: The way of Z: practical programming with formal methods, pp. 31–32. Cambridge University Press, Cambridge (1997)
15. Hayes, I., Flinn, B.: Specification case studies. Prentice Hall International Citeseer, Englewood Cliffs (1987)
16. Bowen, J.: Formal specification and documentation using Z: A case study approach, vol. 66. Citeseer (1996)
17. Hilaire, V., Lauri, F., Gruer, P., Koukam, A., Rodriguez, S.: Formal specification of an immune based agent architecture. Engineering Applications of Artificial Intelligence 23(4), 505–513 (2010)
18. Duke, R., Rose, G.: Formal object-oriented specification using Object-Z. Macmillan, Basingstoke (2000)
19. Abdulrahman, R., Neagu, D., Holton, D.R.W.: Multi Agent System for Historical Information Retrieval from Online Social Network. In: O’Shea, J. (ed.) KES-AMSTA 2011. LNCS, vol. 6682, pp. 54–63. Springer, Heidelberg (2011)

A Multi-agent System for the Modelling of the HIV Infection

Toufik Laroum¹ and Bornia Tighiouart²

¹ University of August 20, 1955, skikda, Algeria
laroumtoufik@yahoo.fr

² University of Badji Mokhtar, Annaba, Algeria
B_tighiouart@yahoo.fr

Abstract. The mathematical tool was used for a long time to model the dynamics of the populations, at present approach of modeling Multi-agents seem to be promising in the views of its capacity to master the complexity of the studied systems. In this work we try to model the population of cells occurring during the infection by the virus of the human immunodeficiency (HIV) to show the efficiency of the approach Multi-agents by comparing with the mathematical approach. The obtained results allowed to bring to light the behaviour and the interactions between the various cells studied in agreement with the biological reports.

Keywords: Simulation Multi-Agents, dynamics of the populations, the infection HIV, the virtual community, bio-informatics.

1 Introduction

The modeling became indispensable for the study of the complex phenomena following the example of the dynamics of the populations. In this sense the mathematical modeling was for a long time the used approach. Recently, the approach of modeling Multi-agents began to be particularly used in the study of the dynamics of the populations relative to the cellular biology so allowing exceeding certain limits of the mathematical approach. This work concerns the study of the dynamics of a population constituted by cells occurring during an infection by the virus HIV. It is the problem which was studied widely mathematically [1] [2].

2 Dynamics of the Populations

The constituent individuals of any population are in a perpetual dynamics where each carried out the life cycle. Consequently all the population evolves in the time and in the space. Today we try to explain biological phenomena to a cellular level, with all the complexity of the interactions between cells, to do it we are going to use the approach Multi-agents which allows modeling the evolution of the phenomenon studied from the elementary behaviour of the various agents.

2.1 Mathematical Modeling

The research in dynamics of the populations is not recent. In 1790 there was a mathematical model of Malthus [3] the exponential growth of a population, then in 1838 the model with logistic growth of Verhulst [4] was proposed. These two models describe the evolution of a homogeneous population, but in 1925 the famous system prey-predator of Lotka-Volterra [5], [6] was the first model describing the evolution of two interacting populations and on which various models were proposed to today. However the mathematical approach possesses certain limits as we try to surmount by the approach multi-agents.

2.2 Modeling by the Multi-agent Approach

The Multi-agents approach is suited well to the study of the complex systems constituted by several entities in interaction. It consists in representing every entity by an agent, then in developing the system with time.

The evolution of different agents with their basic actions and interactions that link will bring out the increase of the phenomenon studied with the appearance of behaviours and unanticipated events [8].

This approach with its low degree of abstraction allows to approach the model in the reality where every agent moves, reproduces, interacts and reacts with the changes of his environment. The most important are that the agents are distinguished some of the others and every agent is marked as could be a natural being because he can be followed at any time during his evolution. So, the addition or the retreat of an agent or of a set of agents is easy [7].

3 Infection by the HIV

An immunizing reaction is mainly expressed by the actions of cells lymphocytes called CD4 and CD8. CD4 lymphocyte produced by the Thymus is responsible for the coordination and the activation among others lymphocytes cytotoxiques CD8. This cell CD4 is subject to the infection by the virus HIV which considers them as an adequate environment to carry out its cycle of proliferation. Thus the destruction of CD4 by the HIV paralyzes the immunizing defense to its source [9].

The phenomenon of the infection by the AIDS virus thus takes place in three stages:

- 1 - Firstly infection: lasts from 3 to 8 weeks is characterized by a fast diminution of lymphocytes CD4 caused by an increase of the viral load.
- 2 - The asymptotic phase: its duration is of around ten years during which the immune system maintains a state of balance between the number of the CD4 and the viral load.
- 3 - It is the phase in which the immune system is depressed because of the fast decrease of lymphocytes CD4 (less of 200 / mm³).

3.1 Mathematical Model

We are interested in this study by the 3D model which is the basic model and the simplest of HIV infection. It is only interested in the evolution of three classes of cells which are CD4 cells (T), CD4 lymphocytes infected by the virus recorded (T*) and finally the virus (V).

The phenomenon is modeled by the following equations [10], where T', T*' and V' indicate respectively the variation rates in density of CD4 cells, infected CD4 cells and virus populations:

$$\begin{cases} T' = s - \delta T - \beta TV \\ T*' = \beta TV - \mu T^* \\ V' = \kappa T^* - cV \end{cases} \tag{1}$$

Table 1. Parameters list of the 3D model

Parameters	Definition
S	Production of CD4 cells by thymus
δ	Mortality rate of CD4 cells
β	Virus infectivity
μ	Mortality rate of infected CD4 cells
κ	Production rate of virus
c	Mortality rate of virus

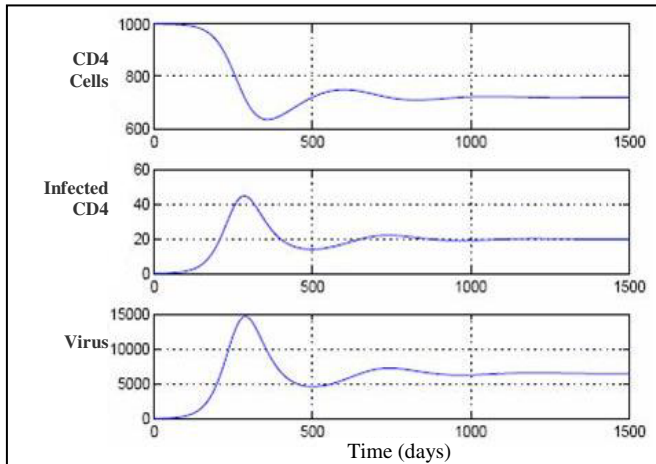


Fig. 1. Results of the mathematical model [9]

Lymphocyte CD4 cells are produced by the thymus with a constant rate equal to S cells in some blood of one mm^3 in a day, and die with a rate of natural mortality equals to δ cells in a day.

The population of lymphocytes CD4 also loses a number of cells which are transformed in infected CD4 cells because of the infection by the virus with a rhythm of βTV where β represents the infectivity of the viruses HIV which is the probability that a meeting between CD4 and HIV is infectious.

The transformation rate of CD4 cells on infected CD4 is the rate of production of the last, dying with a natural mortality rate equal to μ cells per day. An infected CD4 produces a number of viruses with a rate of K Virus HIV a day, these viruses die with a natural mortality rate equals to C virus a day.

This mathematical model allows giving the following results (Fig.1), which represents the phase of firstly infection and the asymptomatic phase in the process of the infection:

3.2 Multi-agents Model

In our case, the objective consists in creating a virtual environment in which various agents evolve and interact between them. It is an environment in 3 dimensions corresponds to 1 mm^3 of the blood.

We used Mad-kit platform (Multi-Agents development kit) to create Three classes of reactive agents feigning the studied cells (The agents cells CD4, the agents infected cells CD4 and the agents virus HIV).

The various agents and the interactions which connect them are represented in the figure Fig.2:

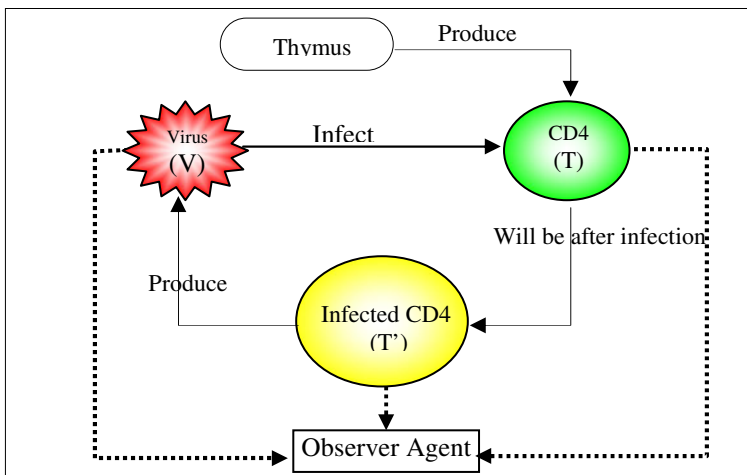


Fig. 2. Interactions in the Multi-agent system

This model Multi-agents is closer to the reality than the mathematical model, which is incapable to express the phenomenon of meeting (contact in the biological sense) between a virus and a CD4 cell.

Effectively, in the mathematical model, the number of the produced infected CD4 agents is calculated by multiplying the total number of the possible meetings between the viruses and the infected CD4 (which is equal to $T*V$) by the parameter β which does not describe faithfully the phenomenon.

In other words, with the mathematical model a population of 100 cells CD4 and 10 viruses gives $100*10=1000$ infected cells CD4 which is not so exact because in the reality this population is going to produce most 10 infected cells CD4 if we suppose that every virus infects one CD4 cell.

That returns because the mathematical approach treats the phenomenon on high-level (consider all the population) contrary to the approach Multi-agents where the treatment is made at the level of the individuals and every meeting is treated (handled) independently of the others, which allows a more exact representation of the reality.

4 Results

4.1 Evolution without Infection

In the absence of the viruses, if we suppose that the thymus produces daily 7 cells CD4, with a mortality rate $\delta=0,007$ that is each possesses a life expectancy (cycle) of

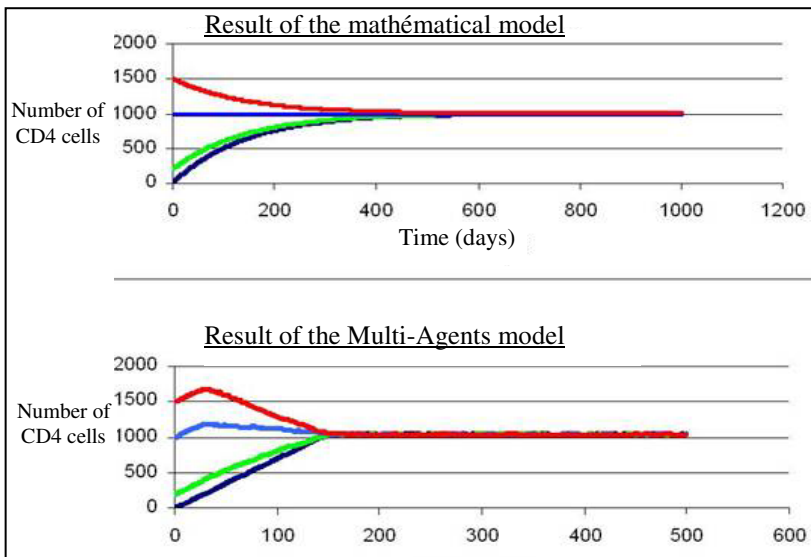


Fig. 3. Evolution of the CD4 without infection

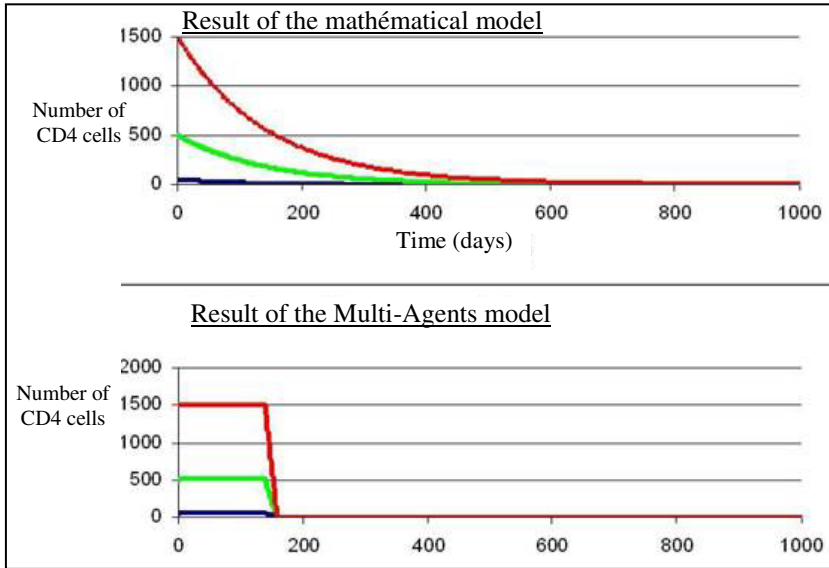


Fig. 4. Extinction of the CD4

$1/\delta=143$ days [9]. We can notice that from several random initial states: 0 cells, 200 cells, 1000 cells and 1500 cells with a random initialisation of the age of cells, the population of the CD4 will converge on 1000 cells CD4 and stabilizes around this value (Fig. 3).

We notice that the model Multi-agents converges more quickly than the mathematical model. The difference between both models also appears in the speed of extinction of a population of isolated CD4 cells that is whose daily rate of production is equals to zero (Fig.4).

The Multi-agents model shows that the population of cells CD4 disappears at once after 143 days (which is the life cycle of cells), and it independently of the initial number of the cells.

4.2 Evolution of the Infection

In an environment which represents 1 mm³ of the blood, evolve three categories of cells: infected CD4, CD4 and the viruses HIV which interact between them by feigning the phenomenon of the infection.

The average life cycles of the agents are defined as follows:

- Agents cells CD4 140 days. With a production rate equals to 7 CD4 every day.
- Agents infected cells CD4: 10 days.
- Agents virus HIV: 3 days.

We notice that the first two phases of the process of the infection are recognizable on the various curves (Fig.6):

- The phase of firstly infection is characterized by a growth of the viral population (initially little numerous) which invaded CD4 cells (initially numerous). The infection of the healthy CD4 gives infected cells CD4 which are going to produce new viruses capable of infecting of the other CD4. This growth persists to reach a rate maximal for which the reduced number of the population of the CD4 becomes a rare resource, consequently a lot of virus die without being able to infect CD4 and producing infected CD4. In that case we notice a fall of the viral load and the number of the CD4 infected.

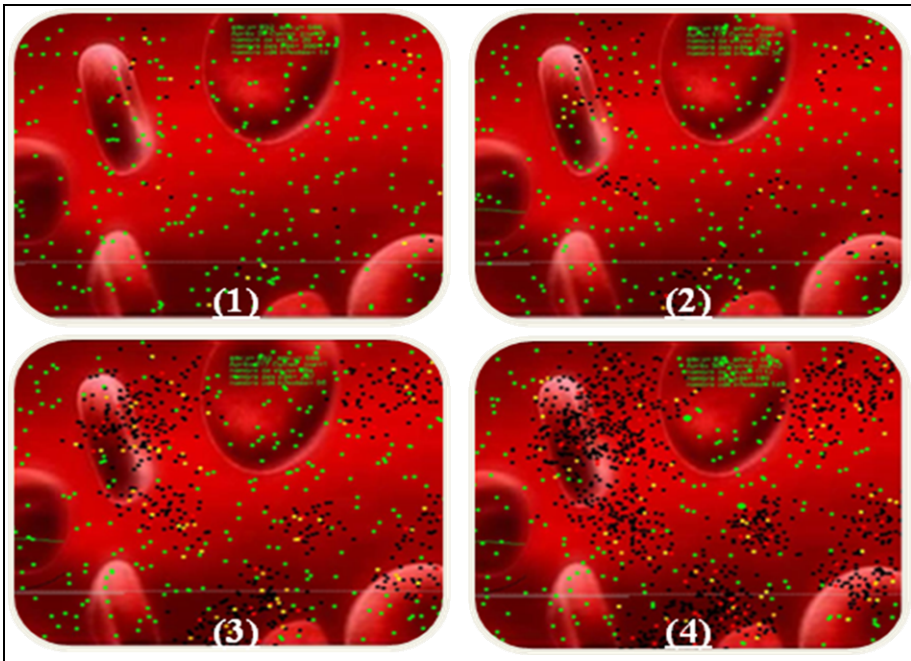


Fig. 5. Propagation of virus in the population of CD4

- The second phase of the infection which is the asymptomatic phase in which a kind of state of balance is established between the rates of the various cells.

The infectivity of the viruses plays an important role on the evolution of the phenomenon of the infection. Indeed, infectivity 100 % leads to an abrupt and fast diminution of the rate of the CD4 with regard to a lower infectivity, and this because the number of viruses produced is bigger (Fig.6).

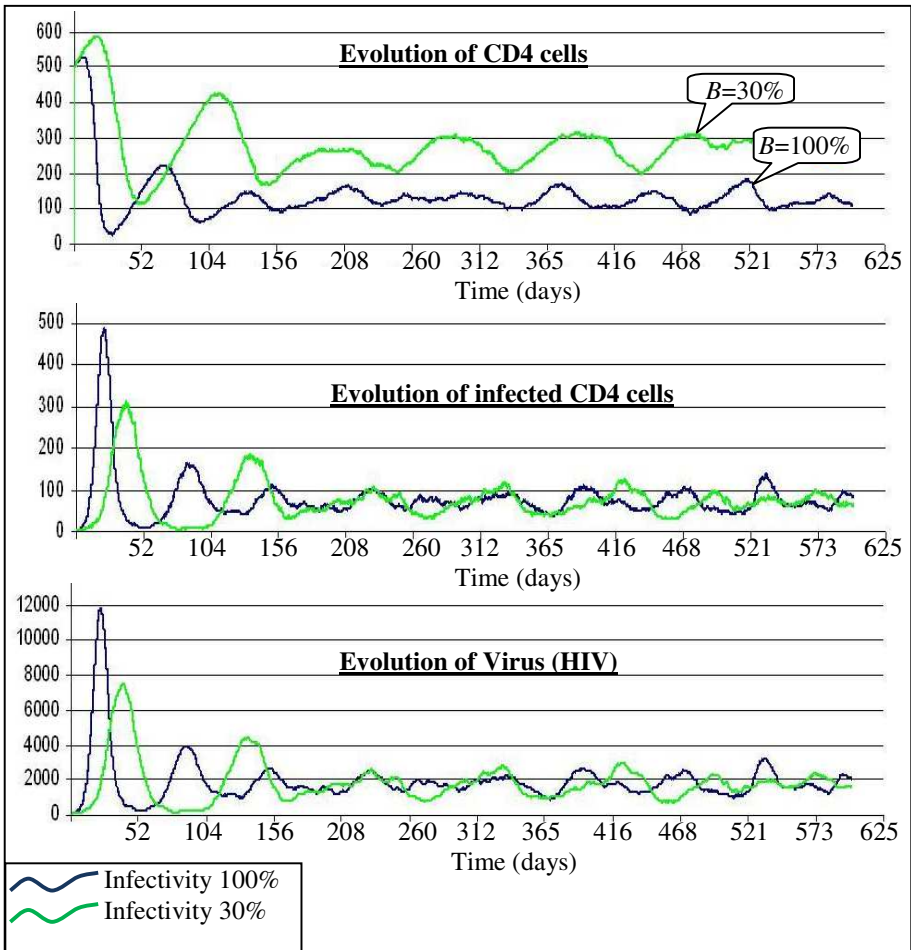


Fig. 6. Results of the Multi-Agents model

5 Conclusion

The population of the agents thus allowed to feign the studied phenomenon by reproducing the behaviour and the interactions between the various studied cells. The approach Multi-agents is suited well to the study of the dynamics of the populations thanks to its capacity to master the complexity of these systems by the direct representation of the individuals. If we define the behaviour of the other particles, the constructed model can reproduce the phenomenon of the infection and will then allow predicting its evolution.

References

1. Moog, C.H., Ouattara, D.A., et al.: Mathematical modelling of HIV infection for an aid in the early diagnosis of therapeutical failures. In: XVI International AIDS Conference, Toronto, Canada (August 2006)
2. Dubois, F. Hervé VJ. Le Meur et Reiss. C.: A mathematical model for HIV dynamics, September 2008.
3. Malthus, T.R.: An essay on the principle of population, 1st edn. J. Johnson in St Paul's Churchyard, London (1798)
4. Verhulst, P.F.: Notice sur la loi que la population suit dans son accroissement, Correspondance mathématique et physique 109, tome X – ou tome II de la 3e série (1838)
5. Lotka, A.J.: Elements of Physical Biology. Williams and Wilkins, Baltimore (1925)
6. Volterra, V.: Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. In: Chapman, R.N. (ed.) Animal Ecology, McGraw-Hill, New York (1926) (1931)
7. Ballet, P.: Intérêts Mutuels des Systèmes Multi-agents et de l'Immunologie. Applications à l'immunologie, l'hématologie et au traitement d'image, thèse de doctorat Université De Bretagne Occidentale (2000)
8. Bouzid, M.: Contribution à la modélisation de l'interaction Agent/Environnement, modélisation stochastique et simulation parallèle, Thèse de doctorat de l'université Henri Poincaré, Nancy 1 (Spécialité informatique). Laboratoire Lorrain de recherche en informatique et ses applications (2001)
9. Ouattara, D.A.: Modélisation de l'infection par le VIH, identification et aide au diagnostic. Thèse de doctorat, Spécialité Automatique et Informatique Appliquée. Université de nantes (2006)
10. Perelson, A.S., Nelson, P.W.: Mathematical analysis of HIV-1 dynamics in vivo. SIAM Review 41(1), 3–44 (1999)

A Multi-agents System for Analysis and Diagnosis of SMEs

Cecilia Zanni-Merk¹, Santiago Almiron², and Dominique Renaud³

¹ LSIIT / FDBT Team (UMR CNRS 7005), Pôle API BP 10413,
67412 Illkirch Cedex, France

² Universidad Nacional de Rosario, Av. Pellegrini 250, S2000BTP Rosario, Argentina

³ LGECO / INSA de Strasbourg, 24 bld de la Victoire, 67084 Strasbourg Cedex, France
merk@unistra.fr, santialmi@hotmail.com, renaudom@free.fr

Abstract. MAEOS is a project about the modelling of the support to the organizational and strategic development of Small and Medium Enterprises (SMEs). Its main objective is to improve the efficiency and performance of business advice to SMEs by automation of these tasks. The main issue is that models coming from management sciences are inherently rich and sometimes contradictory. We are not interested in having a consensus; we want to keep, to its maximum, this plurality, by the highlighting of the contradictory aspects during a diagnosis task. This article proposes our approach to solve this issue.

Keywords: multi-agents system, ontologies, rule-based reasoning.

1 Introduction

MAEOS is a project about the modelling and support of organizational and strategic development of Small and Medium Enterprises (SMEs). Its main objective is to improve the efficiency and performance of business advice to SMEs.

This work aims at establishing a set of methods and software tools for analysis and diagnosis of SMEs. The software tools have to be able to evolve according to the state of the art about SMEs and, in particular, their administrative or legal environments. In addition, they must also be able to reflect the richness and contradictions that are inherent to the models coming from management sciences. Finally, they must permit the consultants to access knowledge coming from diverse sources in an efficient and pertinent way.

In fact, models coming from management sciences are inherently rich and sometimes contradictory. Different authors may recommend contradictory solutions for solving a certain problem and even the same author may change his (her) mind all along his (her) different works. We are not interested in having a consensus; we want to keep, to its maximum, this plurality.

One of the major difficulties in the project is the manipulation of large quantities of knowledge. The volume of knowledge, both theoretical and “expert”, is huge and sometimes, much more detailed than needed. This system consists in a set of knowledge bases whose contents are heterogeneous and this fact raises several issues. Our

approach is to split the different fields of knowledge into smaller modular and homogeneous bases. This approach is supported by the exclusive use of formal ontologies.

The implementation choices were, therefore, directed by the issues highlighted previously: a multi-agents systems or MAS [1] is at the heart of the development, where each agent reasons (following a rule-based approach) on a specific modular ontology. When there is a problem whose nature implies different expertise fields, it is difficult to have a sole expert with enough knowledge in each area and capable to get a good global solution satisfying all the aspects of the problem. It is because of these facts that the choice was to have a set of specialized experts; each of them is an expert on a specific aspect of SMEs management. In this way, each expert in the panel will provide partial solutions on the area where he excels with the final goal of obtaining a global solution. The architecture that is better suited to implement a panel of experts is the blackboard architecture [2], the choice we have made for our implementation. This architecture is easy to update, as the addition or removal of agents are simple operations. Also, as all the agents have the same interface, making data management clear, it is also simple to make changes to the operation of the whole system.

The use of close modular ontologies is possible because the ontologies reflect different points of view on the same reality and because these discrepancies among points of view (usual in companies) only question the logical coherence in certain parts of the model. Arbitration among the different points of view will be, in the end, responsibility of the consultant.

1.1 Ontological Framework of Our Development

The Artificial Intelligence literature contains many definitions of an ontology; many of these contradict each another. For our purposes, an ontology is a formal explicit description of concepts in a domain of discourse (concepts are sometimes called classes), the properties of each concept describing various features and attributes of the concept (also called slots or properties), and the restrictions on slots (sometimes called facets). An ontology together with a set of individual instances of classes constitutes a knowledge base [3]. This modelling approach is called frame-based, and is the one we have chosen for the development of the knowledge-bases of our system.

Reasoning with ontologies and knowledge bases is one of the reasons why a specification needs to be formal one. By reasoning we mean deriving new facts from existing instances in the knowledge base. Therefore, to permit reasoning about instances in the knowledge base, we also need to define a set of rules, usually expressed as logical implications.

In this project, each ontology represents a conceptualization of the knowledge of an expert. In fact, the ontologies we use are out of the traditional development framework because their concepts are not necessarily shared or consensual.

An ontological study was conducted to provide the theoretical foundations necessary for the development. Several ontologies have been studied [4]. Unfortunately, even if these ontologies provide formal exhaustive models that could be applied to our purposes, not many among them include reasoning rules to permit analysis or diagnosis.

It is because of this fact that we needed to develop our own ontologies and rules bases about certain relevant fields for the SME context (organization, production, strategy, finance ...), with the help of experts in the field of Management Sciences.

Four complete ontologies have been developed for the moment: A first one about organization models based on the main works of Henry Mintzberg [5]; a second one about production systems [6] and two more specific ones about SME structure and strategy [7] [8] (Fig. 1). As stated above, these ontologies have been developed following a frame-based approach and implemented with Protégé-Frames[9].

	Mintzberg [5]	Courtois [6]	Reyes [7]	Boissin [8]
Number of concepts	380	379	76	20
Number of relationships among concepts	695	346	145	34
Number of rules	158	54	71	18

Fig. 1. The developed ontologies

To illustrate this article, we will develop here an example about [5] and [6].

2 Architecture of the Multi-agent System

The system is being implemented as a multi-agent system called MAMAS (MAEOS Argumentation with a Multi-Agents System). Each agent reasons with a specific ontology and its set of associated rules. As stated in the introduction, a blackboard is used for communication among agents.

The user (a consultant, for instance) translates his own situation analysis into “as-asserted facts” through the interface.

Each agent has the same behaviour: It examines the blackboard to identify the facts corresponding to concepts it knows. In this case, it launches reasoning about these facts and, in the end, it writes the deduced facts on the blackboard.

Finally, the consultant uses these deductions to organize them into recommendations.

In this system, it is easy to add, replace or remove the different kinds of agents because they can work independently of each other. The agents do not communicate directly with each other but their work on the blackboard may affect the partial solution inferred by other agents. The richness of the final solution given by this system is due to the interconnection of different partial solutions provided by the agents.

This situation implies the production of many pieces of results related to a limited topic. To be well understood by an external user, all the produced results must be combined, at least, by topic. In previous publications [4], we have presented an approach to aggregate results coming from different ontologies. This approach includes a strategy to solve the many limitations resulting from the use of ontologies whose contents are closely related.

2.1 The Blackboard

The main function of the blackboard is to be a passive repository where asserted and inferred facts will be stored. It is not able to modify the information it holds by itself. It is necessary to differentiate the pieces of knowledge that are stored and to have mechanisms to keep track of the changes about the stored knowledge all along the process. It may also implement different management strategies for the access to its content.

The inference engine we are using (Jess [10]) has the possibility to store templates, rules and facts. In our case, the concepts in the ontology will be Jess templates, instances of the concepts will be Jess facts and the new knowledge inference rules will be Jess rules.

On the other side, Jess engines have the possibility of separating the set of templates, rules and facts in what are called “modules”. Modules are a work environment for the ontologies; each module will only store the original and new inferred facts for a given ontology. This architecture ensures that the pieces of knowledge will not be mixed in the same engine, and that each agent will not lose control of the inferred knowledge by his associated ontology. In the same way, the possibility exists to create deliberate relationships among different modules, and this will be helpful for the implementation of the semantic equivalence among concepts in different ontologies (what we call, the “bridges”). In our case, Jess modules help us to store the facts in the blackboard because they simplify the separation and then it ensures a correct retrieval of the pertinent facts by the agents.

In Fig. 2, *Ontology1, Ontology2 ... Ontology_n* represent the modules that are associated to the different ontologies.

We present hereafter an example of the use of modules, using the Jess syntax. Let us consider again our two specific ontologies, the first one about Production Systems [6], and the other about Enterprise Organization [5], the Jess modules will be named, then, PROD and ORG.

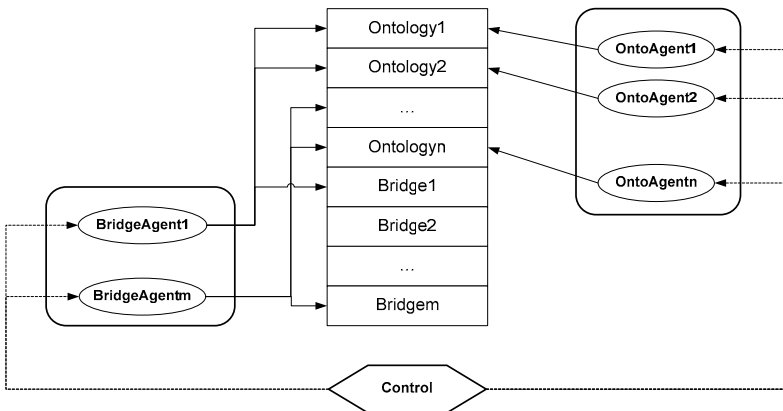


Fig. 2. The MAMAS software

For the Production Ontology, some concepts (or classes) may be written:

```
(deftemplate PROD::Produce_what_is_sold
  (slot Name (type STRING)))
(deftemplate PROD::Reactivity
  (slot Name (type STRING)))
```

An example of a rule:

```
(defrule PROD::Rule_RC16
  (PROD::Produce_what_is_sold)
  =>
  (assert (PROD::Reactivity)))
```

For the Organization ontology, some concepts are:

```
(deftemplate ORG::Speed_of_response
  (slot Evolution (type SYMBOL))
  (slot Name (type STRING)))
(deftemplate ORG::Behaviour_formalization
  (slot Evolution (type SYMBOL))
  (slot Name (type STRING)))
```

A example of a rule for this ontology may read:

```
(defrule ORG::Rule_RM11
  (ORG::Speed_of_response (Evolution Increase))
  =>
  (assert (ORG::Behaviour_formalization (Evolution Increase))))
```

Always in Figure2, Bridge1, Bridge2... Bridgem, are also modules, but they will store the rules to define the semantic equivalences between two ontologies. Although several matching possibilities exist, we have chosen to have a Bridge module for every couple of ontologies where there are semantic equivalences.

The blackboard has two levels of storage. The first level is a pool where the agents store their facts after inference. The second level is the main repository. The control is who decides which facts from the pool will be finally saved in the main repository.

In addition, the blackboard stores the trace of the execution since the first time rules are launched until moment there are no more rules to activate. With this information we will be able to better understand the results and possibly find errors in the rules or initial facts.

2.2 The Agents

There are two kinds of agents: the agents that infer new knowledge from the assertions in the blackboard (we call these agents, `OntoAgent`), and the agents that are

in charge of seeking and defining the semantic equivalences among the different ontologies (*BridgeAgent*). An important design choice is that all the agents have the same interface to their access regardless of the task they perform. This decision facilitates the addition of new kinds of agents.

There are two main methods in the interface of agents, *isReady* and *work*. *isReady* checks the pre-condition of the agent. To do so, it gets the current state of the blackboard and checks if the engine can infer new knowledge. The method *work* gets the current state of the blackboard, infers and then saves the new knowledge in the pool.

There are also specific control routines which implement different executions strategies for the agents.

The main characteristics of *OntoAgents* are the following:

- They only manipulate knowledge from a unique ontology. Therefore, they have an associated module in the blackboard where they store the new pieces of knowledge.
- They are composed of a Jess engine and they convert ontology knowledge (*classes* and *instances*) into the Jess representation (*deftemplates* and *facts*).
- When chosen by the control routines, they retrieve possible new facts from the blackboard. Afterwards, they try to infer internally new knowledge with their own Jess engine. Finally, they store new facts inferred in the blackboard.

BridgeAgents are in charge of exploiting the possible semantic equivalences among the different ontologies. To achieve their goals, they take advantage of the functionalities of the Jess engine; the semantic equivalences are expressed as rules.

To explain their operation, we will use the preceding example. Let us suppose now that there is a semantic equivalence between the *Reactivity* concept in the *Production* ontology and the *Speed_of_response* concepts in the *Organization* ontology. We define, then, a new module to store all these rules and differentiate them from the ones coming from the ontologies with the name *BRIDGE*. This semantic equivalence gives births to symmetric rules (mathematically, the relation between the two concepts is an equivalence relation).

```
(defrule BRIDGE_ORG_PROD::001_Sem_eq_1
  (ORG::Reactivity {Name != Sem_eq_1})
  =>
  (assert (PROD::Speed_of_response (Evolution Increase)
    (Name Sem_eq_1)))
(defrule BRIDGE_ORG_PROD::002_Sem_eq_1
  (PROD::Speed_of_response (Evolution Increase)
    {Name != Sem_eq_1})
  =>
  (assert (ORG::Reactivity (Name Sem_eq_1)))
```

As stated before, agents in this kind of architecture do not work by themselves; they work under the command of the control routines.

2.3 Flow Control Routines

These routines verify which agents have their pre-condition verified (that is, if they are able to begin inference or not) and instructs them when they have to modify the blackboard with new data. These routines implement different algorithms that determine the execution order of the agents. We will present here several strategies we are implementing. One thing is clear, whichever the execution chosen; it has to avoid starvation of the agents, meaning that all the agents will have to be given the opportunity to reason and, eventually, add facts to the blackboard. When the system is initiated, the strategy chosen is set up in these routines.

Different strategies can be implemented for the use of knowledge bases and bridges during the inference process. We have developed three kinds of strategies (sequential, simultaneous and opportunistic), which we will describe below.

- *Sequential*: This strategy aims at introducing a sequential order in the use of the knowledge bases. This order can be identified as a solving process. It depends on the existence of bridges among the ontologies of different knowledge bases (Figure 3-a). Like in any process, iterations can be implemented in the solving process (Figure 3-b). For that, management routines must be added (stop condition, convergence and divergence criteria, etc.). Finally, the implementation of scenarios is conceivable. This requires the creation of control logics. This last proposal is outside the scope of strictly pure sequential strategies.
- *Simultaneous*: The simultaneous strategy aims at using each knowledge base separately and at consolidating the results in a post-processing phase (Figure 3-c). Several strategies can be implemented to consolidate the results: keep the raw results; gather the homogeneous solutions in groups; combine the results [11]; eliminate the contradictions or any mix of these strategies. It is to be remarked that

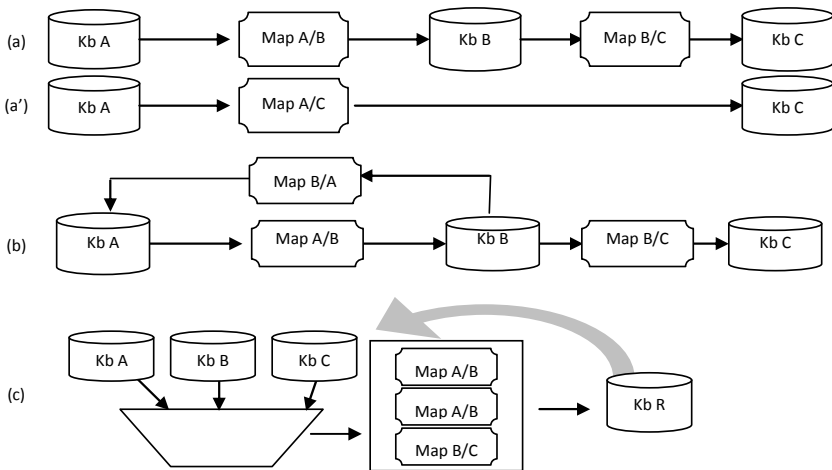


Fig. 3. Examples of strategies

it is also possible to use an iterative process by reinjection of the consolidated or partial results for another round. This also implies the implementation of specific routines (stop condition, convergence / divergence ...).

- *Opportunistic*: The opportunistic strategy is based on the use of knowledge bases with an event trigger process - i.e.: each time that new knowledge is generated, the use of a knowledge base is triggered [12]. In this case, only the policy for managing the blackboard has an influence on the experts' panel. It should be remarked that the stop condition expressed in our previous work was based on a "lack of new knowledge" condition [11]. However other conditions can be implemented.

To have a clear understanding of the impact of each of these strategies, we need to evaluate their influence on the results. We are currently studying these issues to choose pertinent evaluation criteria.

Fig. 4 illustrates a simple example of the use of the *Simultaneous* strategy, only showing one iteration of the system. There are three agents. In this case, only Agent1 and Agent3 can infer. The method `mergeListOfFacts` gets all the new facts and preserve the existing ones.

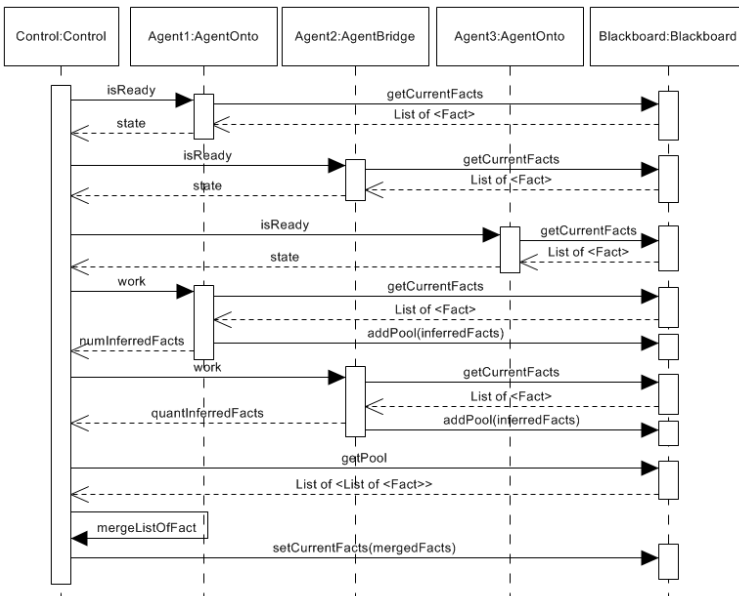


Fig. 4. The use of the Simultaneous strategy

3 Conclusions

In this article, we have presented our first results about a knowledge-based system to assist consultants in their analysis and diagnosis of SMEs. This system consists in a set of knowledge bases whose contents are heterogeneous and this fact raises several issues.

Firstly, there is the management of knowledge structures and backgrounds. It takes the form of complex cycle phases including acquisition, use and maintenance. There is, also, the manipulation of huge quantities of knowledge by the consultant; his expectation is to obtain innovative solutions with several alternatives coming from the system. In addition, as he has to be able to explain his recommendations to his client, he needs to understand the suggestions and analysis provided by the system.

To solve these issues, our approach is to split the different fields of knowledge into smaller modular and homogeneous bases. This approach is supported by the exclusive use of formal ontologies. This point is motivated by the possibilities of manipulation of knowledge and a strict framework imposed by this type of ontology. In this way, we hope to combine the advantages of rigorous knowledge manipulation, adaptability to particular cases and readability of the results.

The main objective of our project is to improve the efficiency and performance of business advice to SMEs and neither to propose solutions to a problem, nor to have the consultant's work done by the software. To do this, the targeted system aims at providing suggestions to help address those areas in which a consultant is less efficient.

Concerning the perspectives of future work, we are exploring different alternatives. As we have stated, several strategies for exploiting the knowledge bases can be used. Further study will permit the analysis of the benefits and limitations of each of them. In addition, it would be interesting to study other strategies coming from the literature about SMAs and the use of blackboards in SMAs. Finally, if this path brings encouraging results, it would be appropriate to use scenarios and / or "meta-strategies". Their goal will be to allow the selection of processing (sequential / simultaneous / opportunistic) and management strategies for the blackboard based on specific criteria or previous cases.

For the moment, the consultants' knowledge is not being considered. We are working on the development of a new kind of agents, capable of reasoning following a case-based approach, to capitalize the knowledge coming from previous studies analyzed by the consultants.

References

- [1] Wooldridge, M.: An introduction to Multi-Agent Systems, 2nd edn. John Wiley & Sons, Chichester (2009)
- [2] Corkill, D.D.: Collaborating Software: Blackboard and Multi-Agent Systems & the Future. In: Proceedings of the International Lisp Conference, New York, USA (2003)
- [3] Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating your First Ontology* (2001), <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>
- [4] Renaud, D., Zanni-Merk, C., Bouché, P., Gartiser, N., Rousselot, F.: A strategy for reasoning with close knowledge bases: Application to knowledge bases on SMEs. In: *Data Mining and Business Intelligence*. Publishing House of the University of Economics, Wroclaw (2010)
- [5] Mintzberg, H.: *The Structuring of Organizations*. Prentice-Hall, Englewood Cliffs (1979) ISBN 0-13-8555270-3
- [6] Courtois, A., Martin-Bonnefous, C., Pillet, M.: *Gestion de Production*. Editions d'organisation (1989) ISBN 2708136488

- [7] Reyes, G.: La moyenne entreprise est-elle spécifique. In: Proceedings of 7ème Congrès International Francophone en Entrepreneuriat et PME, Montpellier, France (2004)
- [8] Boissin, J.P., et al.: Profils de dirigeant et croissance des jeunes entreprises Innovantes. In: Proceedings of Congrès de l'AIMS (2008)
- [9] Hill, E.F.: JESS in Action: Java Rule-Based Systems. Manning Publications, Greenwich (2003)
- [10] Protégé-Frames. Stanford Center for Biomedical Informatics Research (2006), http://protege.stanford.edu/doc/tutorial/get_started/get-started.pdf
- [11] Renaud, D., Zanni-Merk, C., Rousselot, F.: 'A Compound Strategy for Ontologies Combining'. In: Proceedings of KEOD 2009 (2009)
- [12] Bouché, P., Zanni-Merk, C., Gartsier, N., Renaud, D., Rousselot, F.: Reasoning with multiple points of view: A case study. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6279, pp. 32–40. Springer, Heidelberg (2010)

Model of Petri Net for Representing an Agent Plan in Dynamic Environment

Said Brahim^{1,2}, Ramdane Maamri², and Zaidi Sahnoun²

¹ Science computer department, University of Guelma, BP 401, Guelma 24000, Algeria

² LIRE Laboratory, University of Mentouri, BP 325 Constantine25017, Algeria
{brahimi.said, rmaamri, sahnounz}@yahoo.fr

Abstract. In this paper we propose a model of a hierarchical Petri net for representing hierarchical planning. The aim of this model is to allow reasoning in the abstraction level. This property is beneficial for handling the interleaving of planning and execution in centralized or distributed multi-agent planning.

Keywords: Multi-agent planning, hierarchical Petri net, Planning-Execution interleaving, hierarchical planning.

1 Introduction and Motivation

The domains are currently complex, and the emergent applications are characterized by their distributed, dynamic, and unpredictable aspects. The multi-agent systems by their features like the autonomy of the agents, the flexible interaction, the adaptation constitute a good domain of use. For the construction of a multi-agent system evolving in this kind of environment several challenges must be addressed: (i) the environment of the agents can be unpredictable (ii) the failure of the execution of an action can lead to undesirable effects (iii) the goals might be dynamically changed. To face these challenges, the agents forming the system must be able to plan and to monitor their behavior in order to accomplish their goals. For this end each agent must interleave between local planning and coordination. On the other hand, it must interleave between planning and execution. In this context, the definition of adequate models of task and plan constitute one of the major problems [3].

In this paper we are interested to define a model of a local agent plan. The model must be able to take into consideration the concurrent execution of a local plan, and offer the necessary features to monitor the plan's execution. The model proposed is based on a hierarchical Petri net. The motivation of the use of a Petri net is that it is a powerful graphical and mathematical tool for describing the concurrent, asynchronous and dynamic behaviours of an agent. The concept of the marking allows for the description of the evolution state of the plan execution, and the verification of its validation.

Indeed, the Hierarchical Petri net allows the modeling of the hierarchical plan and abstract action. Our model extends the hierarchical Petri net by:

- Associating information to transitions. This information concerns the constraints on the resources used, consumed or produced by the task modeled by transition.

- Defining some rules on abstract transition firing
- Defining Transition marking concept

The first extension consists of incorporating domain-dependent information for the Petri net and defining some rules for propagating this information to an abstract transition. The second extension is inspired by recursive Petri nets proposed in [3]. The abstract transition firing leads to the activation of the sub net that refines the abstract transition. The third extension concerns the marking of the abstract transition to identify the transitions that are in the execution case. The purpose of this extension is to get a flexible and efficient structure that might be used in handling the evolution state of the plan' execution in a dynamic context.

The rest of the paper is structured as follows. In the next section we overview the foundation of the based formalisms. This formalism is based on Petri net. We propose some extension for handling some concepts dependent on the planning domain. This formalism is used for representing the hierarchical plans of the agent. The section tree presents how the proposed formalism includes the domain dependent information. On the other hand we propose how to compute summary information for abstract tasks. In section four we synthesize the benefits aspects of the formalism in the context of local agent planning. In section five we outline some related works. Finally, we conclude the paper by indicating the positive aspects and some perspectives related to future works.

2 Overview of Formalism

2.1 Required Aspects of Formalism

In a dynamic environment the agents plan must take into account the following characteristics:

- At each time the plan must be able to be monitored. The formalism must allow the explicit representation of the state of plan execution. The state that allow verifying the validity and consistency of plan in the current context, and the evolution of resources required by plan execution.
- The possibility to evaluate, at abstract level, properties of the behavior such us liveness and reachability of the goal states.
- Formalism is expected to allow the flexible representing of elastic plans. In dynamic environment some task in plan become useless. Hence the plan must be adapted and update at runtime.

Indeed, the formalism is expected to decrease the complexity and limiting the research space. This property is required for decrease the response time of agent in dynamic situation.

2.2 Model of Hierarchical Petri Net

In this section we illustrate how PN is extended in order to handling abstraction.

Petri Net. A *PN* is defined by the tuple: (P, T, F, W) where: P is a set of place; T is a set disjoint from P of transitions; $F \subseteq T \times P \cup P \times T$ is a set of arc between places

and transitions; $W : F \rightarrow \mathbb{N}$ is weight function of arc. If n is node (place or transition) in PN, $\bullet n$ and $n\bullet$ represent respectively the input and output nodes of n such that $\bullet n = \{n' / (n', n) \in F\}$ and $n\bullet = \{n' / (n, n') \in F\}$. Marked Petri net is defined by (PN, M) such that PN is Petri net and $M: P \rightarrow \mathbb{N}$ is marking function of places. M_0 is the initial marking of Petri net. The marked Petri net represent the state of Petri net. The state might be changed according to the transition firing. A transition t is fireable if and only if: $\forall p \in \bullet t, M(p) \geq W((p, t))$. The firing of transition t leads to change the marked places: $M \leftarrow M + W((t, \cdot)) - W((\cdot, t))$.

Task Petri Net. A Task Petri Net (TPN) is defined by the tuple: (P, T, F, W, s, e) where: (P, T, F, W) is a Petri net; s is a particular place representing the source place ($\bullet s = \emptyset$); e is a particular transition representing the end transition ($e\bullet = \emptyset$); and $\forall a \in F, W(a) = 1$; if $M(s) = 1$ then $\forall p \neq s, M(p) = 0$, if $M(e) = 1$ then $\forall p \neq e, M(p) = 0$; $\forall p \in P, M(p) \in \{0, 1\}$; M_0 is the initial marking in which $M(s) = 1$. The structure of TPN is loop-free and conflict-free. Each transition models one task and a transition firing models the execution of task. We define two particulars TPN , *Sequential-TPN* and *Parallel-TPN* for modeling respectively the sequential and parallel tasks. The *Sequential-TPN* is TPN in which $P = \{s\} \cup \bigcup_{i=1}^m \{p_i\}$, $T = \bigcup_{i=1}^m \{t_i\} \cup \{e\}$, and $F = \{(s, t_1)\} \cup \bigcup_{i=1}^m \{(t_i, p_i)\} \cup \bigcup_{i=1}^{m-1} \{(p_i, t_{i+1})\} \cup \{(p_m, e)\}$. The *Parallel-TPN* is TPN in which $P = \bigcup_{i=1}^m \{p_i, p_i'\} \cup \{s\}$, $T = \bigcup_{i=1}^m \{t_i\} \cup \{f, e\}$, $F = \{(s, f)\} \cup \bigcup_{i=1}^m \{(f, p_i)\} \cup \bigcup_{i=1}^m \{(p_i, t_i)\} \cup \bigcup_{i=1}^m \{(t_i, p_i')\} \cup \bigcup_{i=1}^m \{(p_i', e)\}$, the tasks are connected to source place by a fork transition f .

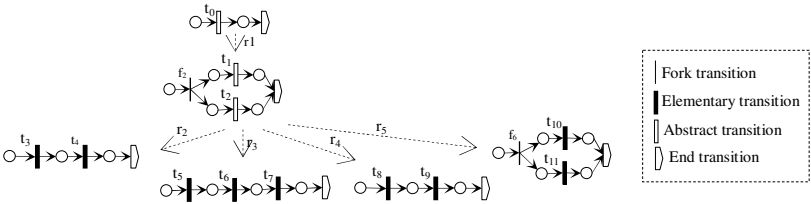


Fig. 1. Example of HTPN

Hierarchical Task Petri Net. A Hierarchical Task Petri Net ($HTPN$) is defined by the tuple: (P, T, F, W, s, e, R) where: (P, T, F, W, s, e) is a TPN in which $T = T_{elem} \cup T_{abs}$ is a set (disjoint union) of elementary and abstract transitions; $R \subseteq T_{abs} \times HTPN$ is a set of refinement rules, each rules $r \in R$ associates to one $t \in T_{abs}$ a refinement $HTPN$. The $HTPN$ has tree-like structure. Each node in this tree has the structure of *Sequential-TPN* or *Parallel-TPN*. A marked $HTPN$ is defined by $(P, T, F, W, s, e, R, Mp)$ where (P, T, F, W, s, e, R) is $HTPN$, $Mp: P \rightarrow \{0, 1\}$ is a marking function of places. The elementary and abstract transition models respectively the atomic and abstract task. The firing of an elementary transition in one $HTPN$ leads to a new marking of the same $HTPN$. However the firing of an abstract transition leads to the activation of one $HTPN$ that might refine this transition. To describe the fireable transition and firing rules, we define so called extended marking of $HTPN$. The extended marked $HTPN$ represents all the activated $HTPN$. It is defined

by tree $Tr = (N, n_0)$ where N is the set of nodes and $n_0 \in N$ is the root of tree. Each node n is a tuple (Pl, Mp, Mt) such that (Pl, Mp) is marked *HTPN*, and $Mt: T_{abs} \rightarrow HTPN \cup \{\varepsilon\}$ is a marking function of abstract transitions. Mt_0 is the initial marking such that $\forall t \neq T_{abs}, Mt(t) = \varepsilon$. n_0 is marked *HTPN* at the very abstract level. A node n' is the child of n in Tr iff $\exists t \in T_{abs}(Pl(n))$ and $\exists r = (t, Pl(n')) \in R(Pl(n))$. In this case $(Mt(n))(t) = Pl(n')$. The initial extended marking is a tree Tr_0 that has the root $n_0 = (Pl, Mp_0, Mt_0)$ only. A transition t is fireable if and only if: $\forall p \in \bullet t, Mp(p) = 1$. Let n be a node in Tr , the firing rule of a transition $t \in T(pl(n))$ is defined according to the type of transition (elementary, abstract, or end) as follow¹:

- If $t \in T_{elem}(pl(n))$: $Mp(n) \leftarrow Mp(n) + (W(Pl(n)))(t, \cdot) - (W(Pl(n)))(\cdot, t)$
- If $t \in T_{abs}(pl(n))$:
 - $Mp(n) \leftarrow Mp(n) - (W(Pl(n)))(\cdot, t)$,
 - $(Mt(n))(t) \leftarrow pl$, Such that $(t, pl) \in R(Pl(n))$, and
 - $N(Tr) \leftarrow N(Tr) \cup \{(pl, Mp_0(pl), Mt_0(pl))\}$
- if $t \in e(Pl(n))$ and n is child of n' :
 - $N(Tr) \leftarrow N(Tr) - \{n\}$
 - $(Mt(n'))(t') \leftarrow \varepsilon$, and
 - $Mp(n') \leftarrow Mp(n') + (W(n'))(t', \cdot)$

2.3 Example

Let Pl be *HTPN* illustrated by figure (Fig. 1). The net are formally defined as follow:

$$\begin{aligned}
 pl_1 &= (P_1, \{t_0, e_1\}, F_1, W_1, s_1, e_1, \{r_1\}), \\
 pl_2 &= (P_2, \{f_2, t_1, t_2, e_2\}, F_2, W_2, s_2, e_2, \{r_2, r_3, r_4, r_5\}) \\
 pl_3 &= (P_3, \{t_3, t_4, e_3\}, F_3, W_3, s_3, e_3, \{\}), \quad pl_4 = (P_4, \{t_5, t_6, e_4\}, F_4, W_4, s_4, e_4, \{\}), \\
 pl_5 &= (P_5, \{t_7, t_8, e_5\}, F_5, W_5, s_5, e_5, \{\}), \quad pl_6 = (P_6, \{f_6, t_9, t_{10}, e_6\}, F_6, W_6, s_6, e_6, \{\}) \\
 \text{where } r_1 &= (t_0, pl_2), \quad r_2 = (t_1, pl_3), \quad r_3 = (t_1, pl_4), \quad r_4 = (t_2, pl_5), \quad r_5 = (t_2, pl_6)
 \end{aligned}$$

In this example pl_1, pl_3, pl_4 and pl_5 are *Sequential-TPN*; the pl_2 and pl_6 are *Parallel-TPN*. In the *HTPN* pl_1 , to refine t_0 , one rule might be used, $r_1 = (t_0, pl_2)$. Perhaps, in the *HTPN* pl_2 , there are four rules that might be applied: r_2 and r_3 to refine t_1 , and r_4 and r_5 to refine t_2 . The *HTPN* pl_3, pl_4, pl_5 , and pl_6 have not an abstract transitions. All transitions in these net are elementary. The Fig 2. Show the sub space representing the extended marking of pl_1 . The $Tr_0 = (\{n_0\}, n_0)$ such that $n_0 = (pl_1, (1,0), (\varepsilon))$ is the initial extended marking of pl_1 . In this case there are one fireable transition, t_0 , because $\bullet t_0 = \{s_1\}$ and $Mp(s_1) = 1$. The starting of the firing of abstract transition t_0 consists of consuming of token of the place s_1 , marking the t_0 by pl_2 , and activation of pl_2 . The new extended marking is $Tr_1 = (\{n_0, n_1\}, n_0)$ such that $n_0 = (pl_1, (0,0), (pl_2))$ and $n_1 = (pl_2, (1,0,0,0,0), (\varepsilon, \varepsilon))$. n_1 is the child of n_0 in Tr_1 . The firing of end-transition e_3 in Tr_6 leads to Tr_7 where pl_3 is cut.

¹ Let $z=(x,y)$ be a tuple, Functions such as $x(z)$ are used to reference the x of z .

3 Plan Model

In this section we are concentrating on the plan model using HTPN. The questions about the handling of resources associated to transitions are also invoked.

Plan-Net. A *plan – net* is HTPN that has the structure of *Sequential-TPN* with one transition. The concrete definition of plan requires the indication of the set of (amount) of resources that might be used, consumed, or produced by tasks in plan at *one time*. Hence, we define a function *Res* for describing the set of these resources. Let *t* be a transition, *Res(t).use*, *Res(t).prod*, and *Res(t).cons* models respectively, the set of consumed, produced and used resources of *t*. These sets have the form $\{(r_1, q_1), \dots, (r_n, q_n)\}$ where r_i is resource name and q_i is amount (or number) of resource r_i . $q_i = (x, y)$ such that x is lower value and y is upper value. These values might be represented by q_i^- and q_i^+ . The lower and upper values come from the fact that abstract transition might be refined by several nets. Each of which might requires some (amount of) resources. In the case of elementary transition, $q_i^- = q_i^+$. The set of resources associated to it is domain-dependent information. However resources specification associated to the abstract transition are inferred from elementary transitions of lowest levels.

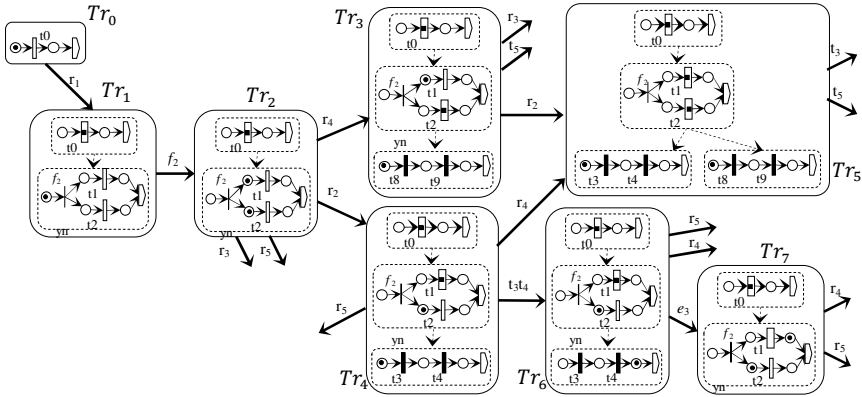


Fig. 2. Marking sub-space of the example presented in Fig 1

3.1 Resource Propagation

The inference of resources specification related to an abstract transition depends on the extended marking and the type of HTPN refining or might refine this transition. The inferring process is based on some propagation procedures. Before to explain the propagation procedures we define the function (*exp*) allowing to compute the expression containing the transitions that have not been fired. Let $n = (Pl, Mp, Mt)$ be a node in the extended marking *Tr*, the function *exp(n)* return the following expression:

- $t_1 < \dots < t_n$ If $Pl(n)$ is *Sequential-TPN* and $(Mp(n)[t_1] \vee (Mt(n))(t_1) \neq \varepsilon)$ and $\bigcup_{i=1}^n \{(t_i, p_i)\} \cup \bigcup_{i=1}^{m-1} \{(p_i, t_{i+1})\} \cup \{(p_m, e)\} \subseteq F(Pl(n))$.
- $t_1 \parallel \dots \parallel t_n$ If $Pl(n)$ is *Parallel-TPN* and $\forall t_{i=1..n}, (Mt(n))(t_i) \neq \varepsilon \vee (\forall p_j \in \bullet t_i, (Mp(n))(p_j) = 1)$

For example in Tr_4 (Fig. 2.) $exp(n_1)=t_1 \parallel t_2$, $exp(n_2)=t_3 < t_4$, $exp(n_0)=t_0$. In Tr_7 , $exp(n_1)=t_2$.

The expression associated to a marked net indicates the transitions from which the resource specification of an abstract transition is computed. The following operators are used to manipulate the resource sets. Let \mathcal{R}_1 and \mathcal{R}_2 be two sets of resources, we define the following operations:

- $\mathcal{R}_1 \boxplus \mathcal{R}_2 = \{(r, q_1 \oplus q_2) / (r, q_1) \in \mathcal{R}_1 \wedge (r, q_2) \in \mathcal{R}_2\}$
such that $q_1 \oplus q_2 = (q_1^- + q_2^-, q_1^+ + q_2^+)$
- $\mathcal{R}_1 \boxtimes \mathcal{R}_2 = \{(r, q_1 \otimes q_2) / (r, q_1) \in \mathcal{R}_1 \wedge (r, q_2) \in \mathcal{R}_2\}$
such that $q_1 \otimes q_2 = (\max(0, q_1^- - q_2^+), \max(0, q_1^+ - q_2^-))$
- $\mathcal{R}_1 \sqcap \mathcal{R}_2 = \{(r, \min(q_1^-, q_2^-), \max(q_1^+, q_2^+)) / (r, q_1) \in \mathcal{R}_1 \wedge (r, q_2) \in \mathcal{R}_2\}$
- $\mathcal{R}_1 \sqsubset \mathcal{R}_2 = \{(r, q_1 \odot q_2) / (r, q_1) \in \mathcal{R}_1 \wedge (r, q_2) \in \mathcal{R}_2\}$
such that $q_1 \odot q_2 = (\max(q_1^-, q_2^-), \max(q_1^+, q_2^+))$

The operators \boxplus , \oplus , \sqcap , \sqsubset , and \odot are commutative and associative. Based on these operators, the propagation rules are defined as follow. Let $n = (pl, Mp, Mt)$ be a node in the extended marking Tr , $t \in T_{abs}(Pl(n))$.

- If $(Mt(n))(t) = pl'$ then $Res(t) = Res(exp((pl', Mp', Mt')))$
- If $(Mt(n))(t) = \varepsilon$ and $\{(t, pl'_i)\}_{i=1..n} \subseteq R(pl)$ then
 - $Res(t).prod = \prod_{i=1}^n (Res(exp(pl'_i, Mp'_0, Mt'_0)).prod)$
 - $Res(t).cons = \prod_{i=1}^n (Res(exp(pl'_i, Mp'_0, Mt'_0)).const)$
 - $Res(t).use = \prod_{i=1}^n (Res(exp(pl'_i, Mp'_0, Mt'_0)).use)$

On the other hand

- $Res(t_1 < \dots < t_n).prod = Res(t_n).prod \boxplus (Res(t_1 < \dots < t_{n-1}).prod \boxtimes Res(t_n).cons)$
- $Res(t_1 < \dots < t_n).cons = Res(t_1).cons \boxplus (Res(t_2 < \dots < t_n).cons \boxtimes Res(t_n).prod)$
- $Res(t_1 < \dots < t_n).use = \sqsubset_{i=1}^n (Res(t_i).use)$
- $Res(t_1 \parallel \dots \parallel t_n).prod = \boxplus_{i=1}^n (Res(t_i).prod)$
- $Res(t_1 \parallel \dots \parallel t_n).cons = \boxplus_{i=1}^n (Res(t_i).cons)$
- $Res(t_1 \parallel \dots \parallel t_n).use = \boxplus_{i=1}^n (Res(t_i).use)$

The information about used, produced, and consumed resources associated to abstract transition give an idea on the execution possibility of this transition. This information is called: *summary information*.

In Fig. 2, the state Tr_3 indicate that t_1 is refined by pl_3 according to the rule r_2 . Suppose that $Res(t_3).cons = \mathcal{R}_1$, $Res(t_3).prod = \mathcal{R}_2$, $Res(t_4).cons = \mathcal{R}_3$, and $Res(t_4).prod = \mathcal{R}_4$, such that $\mathcal{R}_1 = \{(\diamond, 3, 3), (\blacklozenge, 6, 6)\}$, $\mathcal{R}_2 = \{(\diamond, 1, 1), (\nabla, 12, 12)\}$,

$\mathcal{R}_3 = \{(\nabla, 2,2), (\diamond, 4,4)\}$, and $\mathcal{R}_4 = \{(\diamond, 7,7), (\heartsuit, 15,15)\}$. The summary information associated to t_1 are deduced from the transitions of pl_3 as follow. $Res(t_1).prod = Res(exp(pl_3, Mp', Mt')) .prod = Res(t_3 < t_4).prod = Res(t_4).prod \boxplus (Res(t_3).prod \boxtimes Res(t_4).cons) = \mathcal{R}_4 \boxplus (\mathcal{R}_2 \boxtimes \mathcal{R}_3) = \mathcal{R}_4 \boxplus \{(\diamond, 1,1), (\nabla, 10,10)\} = \{(\diamond, 8,8), (\heartsuit, 15, 15), (\nabla, 10,10)\}$. In the case Tr_2 , t_1 is not even refined, there are two rules able to be used to refine it, r_2 and r_3 . Suppose that $Res(t_1).prod = Res(exp(pl_3)) .prod \sqcap Res(exp(pl_4)) .prod = Res(t_3 < t_4).prod \sqcap Res(t_5 < t_6).prod$. If $Res(t_5 < t_6) = \{(\heartsuit, 2, 2), (\nabla, 20,20)\}$ then $Res(t_1).prod = \{(\diamond, 0,8), (\heartsuit, 2, 15), (\nabla, 10,20)\}$.

3.2 Plan Execution

A plan execution is modeled by the evolution of *plan-net*. The transition firing corresponds to the execution of the task modeled by this transition. The marked net gives an idea on how the tasks in plan might be executed and in what order. This description doesn't take in consideration the available resources in some execution context. The execution context represents available resources at one time. It is defined by set $Cxt = \{(r_1, qt_1), \dots, (r_n, qt_n)\}$ where r_i is resource name and qt_i the amount of r_i . We use the expression $qt_{Cxt}(r)$ to describe the quantity of the resource r in Cxt . Hence, the firing of transition depends of the marking state and of the execution context Cxt . An elementary transition t is executable if and only if:

- t is fireable,
- $\forall (r, q) \in Res(t).cons, qt_{Cxt}(r) \geq q$
- $\forall (r, nb) \in Res(t).use, qt_{Cxt}(r) \geq nb$

The evolution of one *plan-net* p is depending of the tree, Tr , representing the extended marking *plan-net*, and the execution context, Cxt . The tuple (Tr, Cxt) represents plan-execution-state of p . The execution of elementary transition t in one plan-execution-state (Tr, Cxt) leads to (Tr', Cxt') such that $Tr[t]Tr'$ and $Cxt' = \{(r_i, q_i - q'_i + q''_i)\}_i$ is the new context such that $(r_i, q_i) \in Cxt$, $(r_i, q'_i) \in Res(t).cons$, and $(r_i, q''_i) \in Res(t).prod$. The execution of elementary transition is modeled by:

$$(Tr, Cxt) \xrightarrow{t} (Tr', Cxt') \quad (1)$$

The execution start of an abstract transition depends of the rule used to refine this transition. We model this as follow:

$$(Tr, Cxt) \xrightarrow{r} (Tr', Cxt) \quad (2)$$

To be able to reason on the execution feasibility of an abstract transition without refinement, we use so called hypothetical execution-state. This state is target state obtained from the execution of abstract transition on one plan-execution-state. Based on the upper and lower amount of resources required (to consume or to use) of an abstract transition t we distinguish two cases:

- The execution of t is *Sure*; if and only if
 - t is fireable

- $\forall (r, q) \in Res(t). cons, qt_{Cxt}(r) \geq q^+$
- $\forall (r, nb) \in Res(t). use, qt_{Cxt}(r) \geq nb^+$
- The execution of t is *possible*; if and only if
 - t is fireable
 - $\forall (r, q) \in Res(t). cons, qt_{Cxt}(r) \in [q^-, q^+]$
 - $\forall (r, nb) \in Res(t). use, qt_{Cxt}(r) \in [nb^-, nb^+]$

In the first case the abstract transition is certainly executable. As there are several alternatives to refine the abstract transition, the execution may lead to several states:

$$(Tr, Cxt) \xRightarrow{t} \{(Tr', Cxt'_i)\}_i \text{ such that } (Tr, Cxt) \xrightarrow{t_{i,1} \dots t_{i,n}} (Tr', Cxt'_i) \quad (3)$$

Where $Tr[t]Tr'$ and Cxt'_i is the new context resulting from the execution of a some elementary transitions sequence $t_{i,1} < \dots < t_{i,n}$. We can abstract the state by abstracting the stat in which the execution context is modeled in term of lower an upper amount of each resource. Hence:

$$(Tr, Cxt) \xrightarrow{t} (Tr', Cxt'') \quad (4)$$

If the execution of abstract transition is possible (not sure) the success of execution is uncertain. In this case the set $\{(Tr', Cxt'_i)\}_i$ might be empty. To raise this ambiguity, we must start the refinement of the transition without changing the execution context as it is described in section 2.2. From this perspective, we define some property about the executability of abstract transition. These properties are: *Executable abstract transition* and *Flexible executability of abstract transition*. Let st be a plan-execution-state, an abstract transition t is executable in st if and only if:

$$st \xRightarrow{t} st' \text{ such that } |st'| \geq 1 \text{ (}|x| \text{ is the cardinality of } x) \quad (5)$$

The executability of abstract transition t is flexible if and only if

$$st \xRightarrow{t} st' \text{ such that } |st'| > 1 \quad (6)$$

These properties might be used for analyzing the execution of a plan in some execution context. Hence the *Executability* of plan-net p is *Feasible* in plan-execution-state st if and only if the transition in abstract level is executable; and it is *Flexible* if the executability of the transition in abstract level is flexible. These two concepts are used to analyze the executability of an abstract plan, without refinement of the abstract transitions, and before start his execution. In a similar manner, we can analyze the success of a plan termination in runtime. The plan that is in execution is represented by the execution state (Tr, Cxt) where $(Mt(n_0(Tr))) (t) \neq \varepsilon$. In this case, the validation of this plan is based on safe termination of the abstract transition in the very abstract level. Based on the upper and lower amount of resources required (to consume or to use) by a marked transition at one time, we distinguish two cases of the termination: *Sure* and *possible*. Let t be a marked transition, Cxt be the current execution context, the execution termination of t is

- *Sure*; if and only if
 - $\forall (r, q) \in Res(t).cons, qt_{Cxt}(r) \geq q^+$
 - $\forall (r, nb) \in Res(t).use, qt_{Cxt}(r) \geq nb^+$
- *possible*; if and only if
 - $\forall (r, q) \in Res(t).cons, qt_{Cxt}(r) \in [q^-, q^+]$
 - $\forall (r, nb) \in Res(t).use, qt_{Cxt}(r) \in [nb^-, nb^+]$

We describe the execution termination of one transition t by the rule:

$$(Tr, Cxt) \stackrel{t}{\triangleright} \{(Tr', Cxt'_i)\}_i \quad (7)$$

Where $Tr = (N, n_0)$, $Tr' = (\{n'_0\}, n'_0)$, and such that $(Mt(n'))(t) = \varepsilon$, $Pl(n) = Pl(n_0)$, $(Mp(n'))(e) = 1$, and Cxt'_i is new context resulting from the execution of a sequence of elementary transitions.

4 Discussion

Our work is based on Hierarchical Petri net. The Hierarchical Petri net is appropriate formalism for modeling the hierarchical plan. The use of Petri net is motivated by the powerful graphical and mathematics tool for describing the concurrent, asynchronous and dynamic behaviors of agent. The concept of the marking allows describing the evolution state of the plan execution. The propagation of constraints about resource (used, produced, and consumed) of an abstract transition allows the reasoning about validity of plan execution at each execution state. This property is benefic for monitoring the execution of plan. It is possible to distinguish between cases in which the plan is executable or flexible. Indeed the Petri net allows model-checking of important property like as reachability, security, liveness, deadlock absence. Finally, by using summary information in abstract transition about used, produced, and consumed resources, we can provide some technique for reasoning on plans interaction.

5 Related Works

Many works have been proposed in the past that are related to ours. In [1], [4], [6] the authors provided some ideas about the propagation of constraints from elementary tasks to abstract ones in order to allow the reasoning on abstract tasks. This work uses a model based on the extension of STRIPS (STanford Research Institute Problem Solver) action representation. The model of the plan adopted in these works is not adequate to represent the evolution of plan execution.

There is another work in [5] that is based on the idea of propagating the information about related resources for each plan. This information is used to select the appropriate (valid, best) plan and to control its execution in some context. They used formalism based on the extension of timed automata. Like our formalism the hybrid automat allows model-checking of important properties like reachability, security, liveness, deadlock absence... however, the local plan has one level of abstraction and the constraints propagation is horizontally computed. The use of one level of abstrac-

tion is not appropriate to repair and adapt the plan. Indeed the automata formalism is not convivial to represent explicitly hierarchical plans and concurrent tasks.

In [3] the authors proposed a formalism called a recursive Petri net. This formalism shares many aspects with ours concerning the transition modeling and firing rules. The proposed model doesn't associate constraints to abstract transitions. Without this information it is not possible to reason about the plans validation and interactions at the abstract level.

The work [7] provides a framework called Petri net plan. The authors proposed a set of operators for handling complex behavior. In this work the plan is represented at one level of abstraction by an ordinary Petri net. The plans must contain all alternatives of behavior. This can present the problem of complexity. On the other hand the plan has rigid structure, thus not being able to dynamically adapt to a new context.

6 Conclusion

In this paper we proposed a model of plan based on Petri net formalism. We extend the hierarchical structure by associated information to transition for handling planning problem. At each time in planning or in execution, the plan might be able to be monitored. The formalism allows the explicit representation of the plan execution state. The concept of executability and flexibility permit to verify the validity and consistency of plan at runtime. This aspect is extremely related to planning and execution interleaving. Also we can at each time of execution make the best decision concerning the choice appropriate alternative for refining abstract transition. The future work will be articulated on the extension of formalism to be able to take in account multi-agent planning.

References

1. Clement, B.J., Durfee, E.H.: Theory for Coordinating Concurrent Hierarchical Planning Agents Using Summary Information. In: Proceedings of the Sixteenth National Conference on Artificial intelligence, pp. 495–502 (1999)
2. El Fallah, S. A. I., Degirmenciyan, C., Marc Modeling, F.: Control and Validation of Multi-Agent Plans in Dynamic Context. In: AAMAS 2004, New York, USA , July 19-23 (2004)
3. El Fallah, S.A., Haddad, S.: A Recursive Model for distributed planning. In: Proceedings of ICMAS 1996. AAAI press, Kyoto (1996)
4. Lotem, A., Nau, D.S.: New Advances in GraphHTN: Identifying Independent Subproblems in Large HTN Domains. In: Proc. Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 14-17, pp. 206–215 (2000)
5. Marc, F., El Fallah-Seghrouchni, A., Degirmenciyan-Cartault, I.: Coordination of complex systems based on multi-agent planning: Application to the aircraft simulation domain. In: Bordini, R.H., Dastani, M.M., Dix, J., El Fallah Seghrouchni, A. (eds.) PROMAS 2004. LNCS (LNAI), vol. 3346, pp. 224–248. Springer, Heidelberg (2005)
6. Thangarajah, J., Padgham, L., Winikoff, M.: Detecting and Avoiding Interference between Goals in Intelligent Agents. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico (2003)
7. Ziparo, V.A., Iocchi, L.: Petri Net Plans. In: Fourth International Workshop on Modeling of Objects, Components, and Agents, MOCA 2006 (2006)

Towards a Predictive Fault Tolerance Approach in Multi-Agent Systems

Samira Telghamti^{1,2} and Ramdane Maamri²

¹ 20 August 1955 University
Skikda, Algeria

telghamti_samira@yahoo.fr

² Lire Laboratory, Mentouri University
Constantine, Algeria

rmaamri@umc.edu.dz

Abstract. Developing a fault-tolerant system require the use of the fault prevention techniques, fault recovery techniques and fault tolerance techniques. In this paper we are going to present a technique which predicts fault occurrence in multi-agent systems. It consists on the identification of critical tasks and the set up of a rule base which defines the situations guiding to break down a running multi-agent system.

Keywords: Agent, Multi-agent system, Fault tolerance, Criticality, Rule base, Replication.

1 Introduction

Fault tolerance problems have been thoroughly studied for distributed systems in general [7]. Recently the multi-agent community has shown interest for this kind of problematic. In order to prevent that a multi-agent system (MAS) stop working properly due to the occurrence of faults, many fault tolerance approaches have been proposed [1] [2] [4] [5] [6]. A robust MAS requires the following techniques [1]:

- Fault-prevention technique: to prevent fault introduction and occurrence;
- Fault-recovery technique: to detect the existence of faults and eliminate them;
- Fault-tolerance technique: to provide services according to the system's objectives in case of faults.

The techniques suggested to solve the problem of faults in MAS are based on the origin of failures. The works [2][4][6] treats those related to the bugs of programming, as for [1] provides a solution to the faults associated to the environment. Others [3] propose a classification of faults in MAS.

In our work we have considered the faults according to their ways of acting on the system. We distinguish the total failure which immediately affects all the agents of the system, and the partial failure which affects a subset of agents or tasks. This latter, will be propagated and becomes a total failure. Whatever their origin, the faults prevent a successful execution of a subset of tasks which are required and should not be

stopped to have the minimum of results expected of the system; i.e. these tasks are necessary so that the system can pass from a state (E_i) to a state (E_{i+1}).

In this paper, we will introduce a first step towards a robust MAS by the proposal for a predictive fault tolerance technique which is based on:

- A new definition of the agents' criticality concept.
- The setting up of a rule base which defines the situations to break down a running multi-agent system.

The paper is organized as follows. Section 2 defines the criticality of the agents. Section 3 describes the predictive fault tolerance strategy. Section 4 presents the fault recovery technique. Section 5 compares our work with related work on fault tolerance techniques. Section 6 concludes this paper.

2 Criticality of the Agents

The replication is an important concept to build a robust system. In our work we have adopting the strategy which consists of first replicate tasks whenever possible, and if not, then to replicate agents [1] but we are taking in consideration the goals of the agents, so that an agent execute only the tasks which are necessary to accomplish its goals. An agent $a_{(i)}$ is critical if it performs at least a critical task. The definition of the critical task depends on the current state of the system defined by the set of agents' plans, as well as, the subset of minimal and necessary goals that the system must satisfy. The algorithm defining the criticality of the agents is presented as follows:

Critical task definition algorithm

Let:

$B = \{b_1, b_2, \dots, b_n\}$: be the set of minimal and necessary goals defined in the system design step;

$T_b = \{t_1, t_2, \dots, t_k\}$: be the set of tasks to satisfy B ;

$T_{plan(i)}$: be a task planned by the agent $a_{(i)}$;

For each plan of agent $a_{(i)}$ do

If $T_{plan(i)} \in T_b$ then

$T_{total} := T_{total} \cup T_{plan(i)}$

End if

End for

The T_{total} set define the tasks in agents' plans to satisfy the B set. It will be subdivided into two subsets:

- **TC**: the set of the critical tasks included in T_{total} , the fault in these tasks does not affect any task in T_{total} . The agents performing at least a TC task will be replicated.
- **TNC**: the set of non-critical tasks included in T_{total} , the fault in these tasks affect the good progress of at least one task of T_{total} . The non-critical tasks will be

replicated in the agents concerned by their results, if the agent does not execute any task in TC.

3 Predictive Fault-Tolerance Technique

We suppose a set of agents $\{a_{(1)}, a_{(2)}, \dots, a_{(n)}\}$ which execute a set of tasks $\{t_{(1)}, t_{(2)}, \dots, t_{(k)}\}$. The execution of a task $T_{(i)}$ depends on a set of pre-conditions and returns a set of results. As these tasks can be joined, the set of the results of the task $T_{(i)}$ (or a part) can constitute the set of pre-conditions of the task $T_{(j)}$.

The predictive technique is based on the set up of a rule base. These rules are generated with logical formulas that define the relationship between the different tasks of the system.

We classified the relationships into two types:

- A fault in the execution of one non-critical task (or several) causes the fault in the execution of one (or several) critical(s) task(s).
- A fault in the execution of one (or several) non-critical task(s) causes the fault in the execution of one (or several) non-critical(s) task(s).

The rule base will be formed by the following rules:

$$F(T_{(i)}) \Rightarrow F(T_{(j)}) \quad (1)$$

Where $F(T_{(x)})$ in the logical formula (1) means the fault in the execution of a task x . The use of the rule base combined with agents' plans allows warning the faults occurrence and their propagation.

The fault tolerance technique we're proposing aims to decrease the number of agents to be replicated. The replication is only applied to the critical agents. The non-critical tasks that affect directly or indirectly the good critical task execution will be replicated in the agents concerned by their results. It permits to decrease the load of communication in the system and preserve also its consistency. The predictive fault tolerance mechanism can be described like this:

Every non critical agent verifies the membership of at least a task $T_{(i)}$ of TNC to its plan. In the positive case, it is going to look for, among the agents which it communicate with, the one that executes at least a task in relation with the task $T_{(i)}$.

The following algorithm describes this mechanism:

Predictive fault tolerance algorithm

BR: The rule base;

$L_{(i)}$: The list of the agents which the agent $a_{(i)}$ has to communicate with;

$LD_{T_{(i)}}$: The group of agents depending on $T_{(i)}$ in their execution;

$tde_{T_{(j)}}$: Estimation of the task $T_{(j)}$ starting time;

$plan_i$: Plan of the agent $a_{(i)}$;

For each task $T_{(i)} \in TNC$ in the plan of agent $a_{(i)}$ do

```

While  $\exists$  at least a rule to activate in BR do
  Determine in  $L_{(i)}$  an agent  $a_{(j)}$  that has planned at
  least a task  $T_{(j)}$  as:
     $F(T_{(i)}) \Rightarrow F(T_{(j)})$ 
    Insert  $(a_{(j)}, T_{(j)}, tde_{T_{(j)}})$  in  $LD_{T_{(i)}}$ 
  End while

If  $LD_{T_{(i)}} = \emptyset$  et  $F(T_{(i)}) \Rightarrow F(T_{(j)})$  /** the case of  $T_{(j)} \in$ 
plani ** /then
   $LD_{T_{(i)}} := LD_{T_{(j)}}$ 
End if
End for

```

This algorithm returns the list of agents which can perform a task $T_{(i)}$ of TNC in case of faults. This agents need the execution results of $T_{(i)}$.

4 Fault Recovery Technique

In this section we are going to present the technique for detection and recovery from failures. It allows a MAS to continue running despite faults. The fault detection of an agent $a_{(i)}$ is based on a mechanism of sending an acknowledgement message in a maximal delay, this delay is defined by the estimation of the minimal time to start the execution of tasks that at least one task $T_{(i)}$ of the agent $a_{(i)}$ depends on.

```

Fault detection algorithm
 $LD_{T_{(i)}}$ : The group of agents depending on  $T_{(i)}$  in their
execution;
 $tde_{T_{(i)}}$ : Estimation of the task  $T_{(i)}$  starting time;
For each agent  $a_{(i)}$  Do
  For each  $LD_{T_{(i)}}$  Do
     $Tmax := \min(tde_{T_{(i)}})$ 
  End for
  timer :=  $tmax$ 
  If timer = 0  $\wedge$   $Msgac(a_{(i)}) = false$  then
     $a_{(i)}$  is in failure
  End if
End For

```

When an agent fails, every non critical task $T_{(i)}$ will be performed by an agent that its execution depends on. The choice of the agent that is going to perform the task $T_{(i)}$,

if there are several agents interested by the results of the task $T_{(i)}$, is done according to some criteria, as: the starting time of the task, the cost of replication and The number of tasks assigned to the agent. When an agent is in failure, each of its tasks in failure will be performed by replica agents. Consequently, the communication links between agents will change, since the agents communicating with the agent in failure will now communicate with some of its replicas. The algorithm demonstrating this mechanism is shown below.

Fault recovery algorithm

$LD_{T_{(i)}}$: The group of agents depending on $T_{(i)}$ in their execution;

$L_{(y)}$: The list of the agents which the agent $a_{(y)}$ has to communicate with;

$Plan_x$: Plan of the agent $a_{(x)}$;

If $a_{(i)}$ is in failure then

For each liste $LD_{T_{(i)}}$ DO

Determine an agent $a_{(x)}$ in $LD_{T_{(i)}}$ to execute $T_{(i)}$

Delete $a_{(x)}$ from $LD_{T_{(i)}}$

For each agent $a_{(y)}$ in $LD_{T_{(i)}}$ DO

Replace $a_{(i)}$ by $a_{(x)}$ in $L_{(y)}$

Replace $a_{(i)}$ by $a_{(x)}$ in $LD_{T_{(y)}}$ /** the case of $(F(T_{(y)}) \Rightarrow F(T_{(i)}))$ **/

End for

If $a_{(x)}$ is not critical and $(LD_{T_{(i)}} = \emptyset)$ and $(F(T_{(i)}) \Rightarrow F(T_{(j)}))$ /** the case of $(T_{(j)} \in plan_x)$ **/ then

$LD_{T_{(i)}} := LD_{T_{(j)}}$

End if

End for

End if

The fault recovery algorithm updates the list of agents which can perform a task $T_{(i)}$.

5 Related Work

There are several works done for building fault-tolerant multi-agent systems such as those based on agent replication [2] [4] [6] or on task and agent replication [1]. The technique in [4] is based on data and computation replication. It allows to automatically and dynamically apply replication mechanisms. The approach described in [6] is based on broker replication; it helps brokers recover from failures. The broker is a

middleware agent that can provide multiple services such as routing requests, search for agents that can perform a given task. In [2] the methodology introduced for replication is based on the concept of agent plan of actions and agent criticality. However, considering the case of a robotic MAS, it is not always possible to replicate costly robots. Consequently, these fault-tolerant techniques could not be applied to a MAS in this case [1].

The task and agent replication technique in [1] addresses the failures which come from the environment. This technique decreases the resulting system complexity by minimizing the number of agents to replicate. However, this strategy defines new sources of failures related to the outside environment which can be identified from the start, during the design phase. Consequently it does not guarantee that the MAS will properly operate. It should be combined with other techniques which deal with programming failures and exceptions such as those described in [2] [4] [6]. This strategy does not take into account the autonomy property on task replication. This property means that an agent take their decisions on their own, by executing only the necessary tasks to accomplish his goal.

Our approach is based on task and agent replication as in [1]. We are taking into account the autonomy of the agent, so that, the task replication is applied on the agents concerned by their results. In our work we deal with faults according to their ways of acting on the system and not according to their origin as in [1] [2] [4] [6]. Whatever their origin, the faults prevent a successful execution of a subset of tasks which are required and should not be stopped to have the minimum of results expected of the system.

6 Conclusion

In this paper, we have defined a new approach for fault tolerance in multi-agent systems. It is based on the set up of a rule base that defines the relation of dependencies between the tasks of the system, and on a new definition of the criticality of the agents concept which is based on the current state of the system, as well as, on the minimal and necessary set of goals that the system must satisfy. Our technique aims to reduce the complexity of the system while minimizing the number of agent to be replicated, and it aims to preserve its consistency because the non-critical tasks will be replicated, only, in the agents concerned by their results. As future work we intend to implement the defined algorithms and to evaluate them through experiment tests.

References

1. Mellouli, S.: A Reorganization Strategy to Build Fault-Tolerant Multi-Agent Systems. In: Kobti, Z., Wu, D. (eds.) *Canadian AI 2007*. LNCS (LNAI), vol. 4509, pp. 61–72. Springer, Heidelberg (2007)
2. De Luna Almeida, A., Aknine, S., Briot, J., Malenfant, J.: Predictive Fault Tolerance in Multi-Agent Systems: a Plan-Based Replication Approach. In: *Proceedings of the 5th international conference on Autonomous agents and multi-agent systems AAMAS 2007*, pp. 921–923 (2007)

3. Potiron, K., Taillibert, P., El Fallah Seghrouchni, A.: A Step Towards Fault Tolerance for Multi-Agent Systems. In: Dastani, M.M., El Fallah Seghrouchni, A., Leite, J., Torroni, P. (eds.) LADS 2007. LNCS (LNAI), vol. 5118, pp. 156–172. Springer, Heidelberg (2008)
4. Guessoum, Z., Faci, N., Briot, J.-P.: Adaptive Replication of Large-Scale Multi-agent Systems – Towards a Fault-Tolerant Multi-agent Platform. In: Garcia, A., Choren, R., Lucena, C., Giorgini, P., Holvoet, T., Romanovsky, A. (eds.) SELMAS 2005. LNCS, vol. 3914, pp. 238–253. Springer, Heidelberg (2006)
5. Ball, E., Butler, M.: Event-B Patterns for Specifying Fault-Tolerance in Multi-agent Interaction. In: Butler, M., Jones, C., Romanovsky, A., Troubitsyna, E. (eds.) Methods, Models and Tools for Fault Tolerance. LNCS, vol. 5454, pp. 104–129. Springer, Heidelberg (2009)
6. Kumar, S., Cohen, P.R.: Towards a fault-tolerant multi-agent system architecture. In: Proceedings of the Fourth International Conference on Autonomous Agents (2000)
7. Birman, K.P. (ed.): Building Secure and Reliable Network Applications. Part III, Reliable Distributed Computing, ch. 12-26 (1996)

Modelling Three-Phase Power Network with a Multi-agent System

Miroslav Prýmek¹, Aleš Horák¹, and Tadeusz Sikora²

¹ Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic
{xprymek,hales}@fi.muni.cz

² Department of Electrical Power Engineering, FEECS,
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava - Poruba, Czech Republic
tadeusz.sikora@vsb.cz

Abstract. Multi-agent systems have already proven to be useful for modelling processes in distributed environments, such as an electricity distribution network. The autonomous design of each network element and its capabilities allows to adapt the model to any situation and the results can be used for monitoring as well as real time control.

One disadvantage of such models lies in a seeming inability to evaluate global analytical computations. In this paper, we show how such computations can be performed within a power network simulation system by means of purely local interactions and an iterative communication flow. The results are then compared to a standard analytical approach.

Keywords: power network simulation, power network modelling, multi-agent simulation.

1 Introduction

Static power network models are often used for *global analytical computations*^[1] of dynamic processes between the network elements [1–3]. Multi-agent systems (MAS) offer new perspectives in computing complex network models. The standard analytical approach is suitable for optimizations, but requires usually a full recalculation after any (even slight) modifications to the network logic. To be able to gain full power of MAS power network modelling, the computation algorithm should not be based on global data and should not use any analytical technique that takes into account all network elements.

In the following text, we describe the algorithm that is based purely on *local interactions*, it is referred to as the “*local approach*” to power network modelling. This algorithm works only with neighbouring elements of the power network as “agents” in MAS (usually a *power line agent* and a *source/consumer agent*).

¹ In this perspective, “*global*” refers to the fact that the computations need to work with the network data as a whole, usually a (multi)graph.

The algorithm is included and modelled in the MAS power network simulation system named Rice [4].

The algorithm was tested on a model of a real-world power network. The network steady-state was computed in two systems – *PAS Daisy* [5, 6], as an example of a stable and widely used system for power network modelling with the global analytical approach, and in Rice. The results of the modelled networks are presented further in this text.

2 Real Power Network Model for Power Network Steady-State Computation

For testing purposes of the Rice MAS the 10 kV electric distribution network (EDN) was chosen. Both the network parameters and load data were acquired from a real electric network. The network is fed from one 110 kV/10 kV transforming station named MART.

The network and the loads are typical for urban area. In the test system there are two big loads (Fnem, Energ), where the outgoing feeders from the supply node MART can be joined. Nevertheless in this test case the interconnection was not taken into account and the tie lines K5 and K10 are switched off.

As it is first design of MAS calculation of EDN steady state, only direct-current model was considered. So only resistance of lines was taken into account.

The line and node parameters are in Table 1. The power network scheme is depicted in Figure 1.

Table 1. Node and branch parameters and calculation results

Name	Load (kW)	Voltage (kV)	Name	Length (m)	Resistance (Ohm)	Current (A)
Mart	Source	10	K1	811	0.1	74.70
BudN	220	9.8905	K2	811	0.1	66.40
BytV	250	9.9743	K3	811	0.1	49.80
CSAV	150	9.9441	K4	811	0.1	29.05
Dom_D	200	9.6100	K5	811	0.1	0
Energ	1100	9.9594	K6	1169	0.15	211.65
FNem	3000	9.7940	K7	1169	0.15	203.35
Menz	470	9.9602	K8	1169	0.15	190.90
Prad	260	9.8403	K9	1169	0.15	178.45
VTP1	400	9.9651	K10	930	0.12	0
			K12	2855	0.37	62.25

3 Analytical Computation of the Steady-State

To compare the results of steady-state MAS calculation in Rice the steady state of the test system was calculated also in program PAS Daisy Off-Line, which is

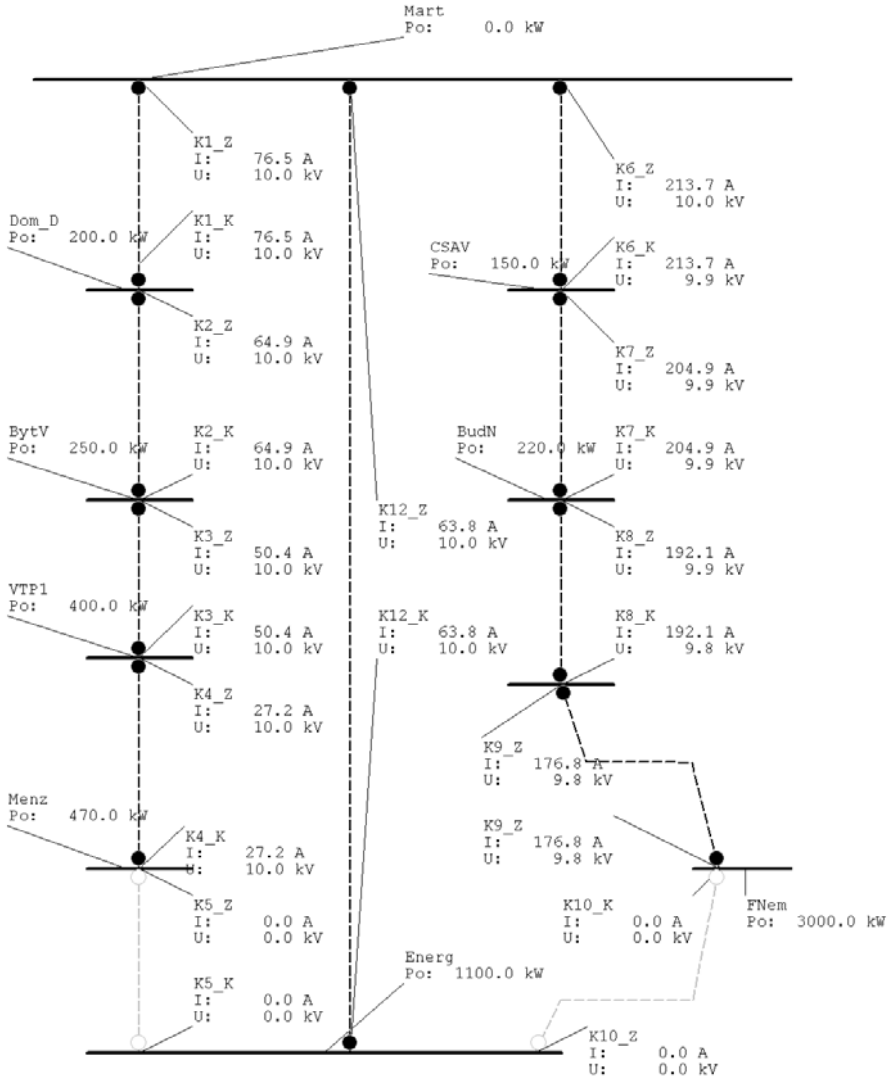


Fig. 1. Scheme of the evaluated power network

a part of the PAS Daisy system [5, 6]. This program is designed for calculating electric network modes, and it is commonly used by the utilities for distribution system operating and planning. The network parameters used for computation in PAS Daisy come from the model described in Section 2.

The analytical calculation is based on modified Newton-Raphson iteration method [7] for fast and reliable convergence of calculation. In the employed method it is no need to build and calculate Jacobian at each iteration step.

According to accuracy required, and network state, the calculation ends within 3 to 6 iterations.

The PAS Daisy program calculates with constant power loads, independent on voltage change (e.g. voltage drop).

4 MAS Computation of the Steady-State in Rice

As mentioned above, the multi-agent approach motivates developers to implement completely independent software components. The centrally-controlled functionalities are kept to the necessary minimum. Even the topology of the network is implemented as an *agreement* between agents. The main advantage of this approach is that the network topology can be changed dynamically according to the observations and decisions of the agents.

The agents which cause the first step to establish the topology, are the ones representing the power lines. For each agent there are two pieces of knowledge inserted into their knowledge base: the name of the agent representing the line's input and the name of the output agent. Immediately after the launch of the simulator, the line will send these agents a message telling "I am your input" and "I am your output." All the actions related to establishing a power flow channel will be taken as a reaction to the arrival of these messages as will be illustrated in the following paragraphs.

The modelled is 10 kV (nominal line-to-line voltage) three phase alternated current EDN. But in a matter of design simplicity, the network is considered ideally symmetric and thus only one phase is calculated. An other simplification is used, as the model is calculated as direct current power flow, so the lines have resistance R only, and the load power factor is equal to 1.

Rice computes the one-phase model first (line-to-ground voltage, 1-phase power) and then the values are transformed to 3-phase results, which are presented to users (line-to-line voltage, 3-phase power). In the following text, the described agent actions thus work with 1-phase values. The 3-phase values are calculated in consequence of the 1-phase ones.

The Source Node. The source node is an input point of the network. Its voltage is U_{in} . The immediate overall consumption of the network connected to this source is:

$$P = U \cdot I \quad (1)$$

The value of the maximal amount of electric power the agent can deliver to the network can be defined as a pre-set value. All power-increase requests will be denied if they exceed this limit.

The source node agent knowledge base contains the following items:

1. U – voltage (Volts)
2. I – current (Amperes)
3. P – load (Watts) – computed according to equation (1)

4. P_max – maximum load allowed
5. *overloaded* – true/false, meaning that the source is actually overloaded

To make all the knowledge pieces correct and make the agent cooperate with other agents in the network, we must define these behavioral reactive primitives:

1. an agent tells me that it is connecting to my output \longrightarrow subscribe for its I_demand knowledge
2. subscribed I_demand changes \longrightarrow change the current I accordingly
3. the current I changes \longrightarrow recalculate the value of the load P
4. the voltage U changes \longrightarrow recalculate the value of the load P
5. if the load $P > P_max$ \longrightarrow assigns **true** to *overloaded*
6. if the load $P \leq P_max$ \longrightarrow assigns **false** to *overloaded*

The Power Line. The power line agents initialize the network topology establishment. Each such agent automatically sends the only two unsolicited messages in the system thus interconnecting and starting the whole simulation. At each moment the line is connected to one source node and one consumer node at most but these connections can dynamically change throughout the simulation.

A power line agent has some features in common with the source node. Besides them, the power line has one specific piece of knowledge R , which is the *resistance* of the line, and two voltage indicators: the *input voltage* U and the *output voltage* U_{out} .

The line resistance causes a voltage drop on the line. The voltage drop is specified as:

$$\Delta U = R \cdot I \quad (2)$$

The output voltage is reduced by the voltage drop:

$$U_{out} = U - \Delta U \quad (3)$$

The behavior primitives of the line are as follows:

1. simulation start
 - \longrightarrow send notifications to your input and output
 - \longrightarrow subscribe for the voltage U and the current I of the input
 - \longrightarrow subscribe for I_demand of the output
2. the voltage U has changed \longrightarrow recalculate the output voltage U_{out} according to equation (3)
3. the current I has changed \longrightarrow recalculate the output voltage U_{out} according to equation (3)

Details of the agent communication phases are discussed below.

The Consumer Node. The consumer agent represents one of the endpoints in the power distribution network. It initializes the flow of the electric power according to its current power consumption demands. This is implemented by setting the I_demand knowledge (which is automatically subscribed when the line connects as an input to this agent) according to the P_demand knowledge and equation (1).

The pieces of knowledge of the consumer agent are:

1. I – the input current granted from the Source Node through the network
2. U – the input voltage that propagates from the Source Node
3. P_demand – the actual power consumption requested by the Consumer Node
4. P – the actual power delivered by input (calculated using the equation (II))
5. $P_threshold = P - P_demand$ – the desired sensitivity of the computation. When P and P_demand differers less than $P_threshold$ the Consumer Node demand is saturated.
6. I_demand – the current corresponding to P_demand

The consumer agent's behavioral primitives are:

1. generate P_demand according to the time of day and the given consumption table
2. the power consumption P_demand has changed \longrightarrow change the corresponding current I_demand according to equation (II)
3. I has changed \longrightarrow consider if the power demand is saturated
4. the consumption is not saturated \longrightarrow increase I_demand
5. the predefined consumption function

The main phases of the simulation are as follows – the first three phases represent the system initialization. The energy flow phase is started by setting P_demand knowledge of some agent to a non-zero value. The consumer node uses the network with the established topology to initiate the actual energy flow throughout the network. When P_demand is set to a non-zero value, I_demand is computed and dispatched to the subscriber (input line). The request then propagates throughout the whole network to the power source which decides if the request can be satisfied. If yes, it sets the I knowledge which then propagates through the network back to the consumer. All voltage drops on lines and the current splitting are computed automatically by the agents representing the appropriate network elements.

Throughout the whole process, the consumer agent simulates the power demand variations in time. All the power flow indicators get recomputed automatically. The Rice network power flow model implementation is based completely on local interactions between the agents and on-demand knowledge propagation. No information is communicated without being requested. Since each part of the simulation process is based on changes of agents' knowledge and the reactive nature of the agents, it is possible to manually change any of the involved indicators interactively during the simulation, including the network topology.

4.1 Comparison of the Two Steady-State Computation Approaches

A comparison between analytical calculation (PAS Daisy) and MAS calculation was performed. In this comparison, we have concentrated to the analysis of the results obtained by both methods. The size of the modelled network did not allow to make exact and meaningful measurements of the computational time of the methods – both computations were performed in fractions of seconds.

Table 2. Values and value differences measured during the comparison. Absolute error values are displayed in permille (per thousand) values ‰.

	PAS Daisy I (A)	Rice #1 I (A)	Rice #2 I (A)	Err #1 (abs.‰)	Err #2 (abs.‰)
K1	76.50	76.28	76.73	2.82	2.98
K2	64.90	64.73	64.98	2.56	1.29
K3	50.40	50.30	50.42	1.98	0.36
K4	27.20	27.16	27.24	1.65	1.60
K6	213.70	213.43	213.66	1.24	0.18
K9	176.80	176.83	176.84	0.20	0.23
K8	192.10	192.07	192.10	0.16	0.02
K12	63.80	63.77	63.77	0.52	0.52
K7	204.94	204.77	204.95	0.82	0.06

Table 3. Voltage results comparison (errors in permille ‰ values)

	Rice	PAS Daisy	Err (abs.‰)	Rice	PAS Daisy	Err (abs.‰)
	Voltage U (V)			Voltage drop ΔU (V)		
BudN	9890.79	9890.52	0.0268	109.21	109.48	2.42
BytV	9974.44	9974.29	0.0152	25.56	25.71	5.92
DomD	9986.17	9986.1	0.0072	13.83	13.90	5.20
CSAV	9944.26	9944.12	0.0139	55.74	55.88	2.49
Energ	9959.32	9959.36	0.0042	40.68	40.64	1.03
FNem	9794.45	9794.03	0.0428	205.55	205.97	2.04
Menz	9960.41	9960.17	0.0237	39.59	39.83	5.94
Prad	9840.63	9840.28	0.0353	159.37	159.72	2.18
VTP1	9965.33	9965.13	0.0198	34.67	34.87	5.66

The results of line current calculation are presented in Table 2. Two calculations in Rice were performed: Rice #1 with low sensitivity of consumer agents to the $P - P_{demand}$ difference, and Rice #2 with higher sensitivity (smaller $P_{threshold}$). The agents with higher sensitivity in test Rice #2 performed more iterations and their error value (Err) is mostly lower, and also closer to the analytical results. For the error calculation the analytical values (PAS Daisy) were taken as a reference. Higher sensitivity has shown generally better accuracy of the MAS calculation.

Table 3 shows Rice and PAS Daisy voltage differences in nodes. The voltage difference calculated as a proportion is very low even when expressed in permille (per thousand) values. This fact is caused by low voltage drop and thus small differences on relative scale between voltages calculated by Rice and PAS Daisy. We have thus displayed also the calculated voltage drops in Table 3. The PAS Daisy results were taken here as a reference. The difference varies from 1 to 6 ‰ in absolute permille values.

As a result of this comparison, we can tell that the average difference between MAS and analytical computations were less than 1‰ in current and about 2‰ in voltage drops (less than 0.02‰ in voltage values). We believe that in this respect the MAS results can be regarded as equivalent to the analytical ones while keeping the distributed manner of the overall computational model.

5 Conclusions

In this paper, we have described the local approach to computing power network steady-state by means of a model in a multi-agent system (MAS) named Rice. Using such local computations offers best results in the network flexibility and distributivity, making it an ideal tool not only for modelling, but also for monitoring and control.

To verify the algorithm output, the computation was modelled in two systems – PAS Daisy as a global approach system and Rice MAS as a local approach system. The results of these two computations were discussed and they prove that the iterative local approach provides practically the same values as the global analytical modelling, while keeping the flexibility of MAS.

Acknowledgments

This work has been partly supported by the Czech Science Foundation under the project 102/09/1842.

References

1. Hcbson, E.: Network constrained reactive power control using linear programming. IEEE Transactions on Power Apparatus and Systems PAS-99(3), 868–877 (1980)
2. Schweppe, F.C.: Power system static-state estimation, Part III: Implementation. IEEE Transactions on Power Apparatus and Systems, 130–135 (1970)
3. Krishnamoorthy, S., Chowdhury, M.H.: Investigation and a practical compact network model of thermal stress in integrated circuits. Integrated Computer-Aided Engineering 16(2), 131–140 (2009)
4. Prýmek, M., Horák, A.: Multi-agent Approach to Power Distribution Network Modelling. Integrated Computer-Aided Engineering 17(4), 291–304 (2010)
5. Střída, F., Stacho, B., Rusek, S.: Network steady-state modelling in the Bizon projektant program. In: 10th International Scientific Conference Electric Power Engineering (EPE 2009), Ostrava, Czech Republic, VŠB TU Ostrava, pp. 186–189 (2009)
6. DAISY, s.r.o.: PAS DAISY Bizon Projektant (2010), <http://www.daisy.cz>
7. Kundur, P., Balu, N.J., Lauby, M.G.: Power system stability and control. McGraw-Hill Professional, New York (1994)

A Method of User Modeling and Relevance Simulation in Document Retrieval Systems

Bernadetta Mianowska and Ngoc Thanh Nguyen

Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Bernadetta.Mianowska@pwr.wroc.pl, Ngoc-Thanh.Nguyen@pwr.edu.pl

Abstract. Modeling users' information interests and needs is one of the most important task in the personalization of information retrieval domain. Search engines' algorithms are constantly improved but they still return a lot of irrelevant documents. For this reason a user is not able to quickly find a document with necessary information. In this paper we propose a user profile that is updated based on novel relevance judgment method. Proposed algorithms are a part of personalization agent system. As performed experimental evaluations have shown, the distance between current user profile and user preferences is decreasing when our algorithm is applied.

Keywords: user profile adaptation, simulation of user behaviour.

1 Introduction

Modeling user information needs is nowadays the most important task in information retrieval and personalization research area. Information overload and disorder are mentioned as the most often reasons for the lack of user satisfaction in information searching process [6]. Any advanced method of information retrieval will be not sufficient if user does not enter a few words into the browser. The user can not be obligated to construct a query that will exactly express his information needs. Among the reasons of this fact one can mention: lack of user's knowledge or difficulties in formulating a query, user being unaware of real information needs or simply user's laziness - when he would rather look through longer list of results than formulate maybe more complicated but proper query.

There are many methods that can improve user request based on some additional knowledge about user, his interests, previous activity in the system or query context. Popular algorithms of queries extension for disambiguating the meaning of user's query are presented by Bhogal et al in [1]. Information about the user interests is stored in user profile. The most important issue in each personalization system is updating process because only based on current information about the user, the system can work properly. To gather such information the agent-based system needs to attentively observe which documents were useful for the user. Two methods are applied. A simple personalization agent-based system asks explicitly which documents were useful for the user. It is a very comfortable method (for the system) but usually

user would not like to spend a time marking if a document was relevant or not. More sophisticated systems try to guess whether document was useful for the user based on some observations of user's activity. Relevance degree depends on the questions: does user open, save or print the retrieved document (was it a useful document), or rather the document has been opened and almost immediately closed (document was not useful for the user) [5].

Personalization system, as any system that interacts with the user needs some training data, which is used to learn users' preferences and penchants. When the user model is built, document relevance can be established. It is an important thing in document recommendation to the user and in profile updating process because relevance document has positive influence on user profile, while non-relevant documents can be a negative feedback for the model [10].

In this work we present a user profile and its adaptation method in specified situation. We assume that we do not have a real user that will submit series of queries and mark relevant documents. We are simulating user's behaviour in information retrieval process and we propose of relevance judgment method based only on simulated user preferences. The main advantage of user simulation is saving time but such approach needs to regard many intuitive assumptions how user behaves in real situation. Proposed user model and its adaptation method are used in the Personalization Module of Agent-based Personal Assistant for Internet users that was first developed in our previous work [9].

This paper is organized as follows: Section 2 contains an overview of personalization methods and techniques to calculate document relevance. Section 3 presents user profile structure and document representation. In Section 4 we describe how user behaviour and document relevance is simulated in searching process. Section 5 presents our experiment and discussion about the results. Conclusions and future works are contained in Section 6.

2 Related Works

User's information need expressed in the query form usually is different then the sequence of words entered as a query. All relevance definitions say about the dependency between document and users' needs. The query entered by the user is only the tip of what the user wants.

According to Robertson et al [12] document is relevant when the user wants it and this document is relative to his search for information. If user does not want this document, they call it "non-relevant". Rijsbergen [11] also emphasizes that relevance is a subjective notion. Different users may differ about the relevance or non-relevance of particular documents to given questions.

Wallis and Thom [15] claim that intuitively a relevant document is one that satisfied some requirements in the user's mind. They also suggest that it is brave assumption that requester (user) knows what is relevant because what a user wants may not be what he describes. Moreover, in many cases users do not express their desires clearly and the same query submitted by different users can have significantly different meanings.

According to Crestani [2] relevance is a relationship that may or may not hold between a document and a user of the IR system who is searching for some information: if the user wants the document in question, then we say that the relationship holds. With reference to the preceding model, relevance is a relationship between a document and user's information. If the user wants the document d_i in relation to his information need q_k , then d_i is relevant.

There exist many probabilistic models of information retrieval that have been applied for years. Most of them meet the conditions assumed by Rijsbergens' two principles of building document rankings [11]:

1. The document's relevance to a request is independent of other documents in the collection.
2. If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.

Latent Semantic Indexing (LSI) is also very popular and effective method for determining if a document is relevant to the user based on a few words submitted by user [13]. LSI uses information about synonyms and context that the selected term is occurring in the text. Unfortunately, this method requires additional knowledge about statistical properties of documents' set that is not always available. Moreover, this model of document relevance assumes the relationship between a user query and document but not between document and real user information needs.

However, the most effective approach of relevance judgment can be simply asking a user if presented document was relevant, this method is very uncomfortable for the user. As Sugiyama et al shows in [14] we can not believe that user ratings will be permanent with time or will be useful for other users. Therefore, search system should directly and exactly capture the changes in each user's preferences without any user effort in order to provide more relevant information for each user.

Also most probabilistic approaches have only addressed the relation of query to document (term matching), not the user's engagement with the document in relation to their problem, i.e. in many cases a ranked document of 0.80 (80% relevant to the query) can be more relevant to a user's information problem at hand than one ranked 1.00 (100%) [4].

In this context the following question seems to be reasonable: how to establish or calculate the relevance between user information needs and retrieved document? We would try to answer this question in the next sections and evaluate proposed algorithm based on particular database of books' descriptions.

3 User Profile and Document Representation

In this section we present our approach to user modeling for an agent-based document retrieval system and method of keeping the user profile up-to-date.

In our previous works [8] we proposed hierarchical structure to present user profile and dependencies between terms in it (relationship "is-a" taken from WordNet

ontology [17]). In this work the hierarchical structure is not adequate because we do not have information about relationships between terms directly from used database.

We have a large set of documents - books from Main Library and Scientific Information Centre in Wroclaw University of Technology [16], each of them is described by its title and a set of complex terms (here called ‘appellations’):

$$d = (tit, \{a_1, a_2, \dots, a_L\}). \tag{1}$$

where *tit* is the title of the book and *L* is the number of appellations for considered document and each appellation is constructed as follows:

$$a_l = (a_m^l, a_{wg}^l, a_{wh}^l, a_s^l); l \in \{1, L\}. \tag{2}$$

and $a_m^l, a_{wg}^l, a_{wh}^l, a_s^l$ are respectively: the main term (with different wording but with the same meaning), sets of broader terms, narrower terms and correlated terms (“see also”). The selected document may have some of these terms undefined. An exemplary document is described as follows (database is in polish; here only translation is presented):

title: Classical and fuzzy databases: models, queries and summaries
 authors: Krzysztof Myszkorowski, Sławomir Zadrozny, Piotr S. Szczepaniak.
 description: *Database - management*
Fuzzy logic - application

“*Database*” and „*Fuzzy logic*” are basic complex terms (some additional specification is added like “*application*” and “*management*” to specify that this book is about management of database and application of fuzzy logic). The appellations set “*Fuzzy logic*” has the following form:

- $a_m = \{\text{Fuzzy logic, Logic of uncertainty, Logic of fuzzy sets}\};$
- $a_{wg} = \{\text{Mathematical logic, Multivalued logic}\};$
- $a_{wh} = \{\text{Fuzzy systems}\};$
- $a_s = \{\text{Theory of probability, Fuzzy control, Fuzzy sets}\}.$

From the set of appellations a structure of terms could be extracted but it would not be a deep hierarchy and the cost of its constructing would be greater than possible benefits from using structure. In this situation, user profile does not contain any information about relationships between terms and the following simple structure is proposed:

$$UP = \{(t_1, w_1), (t_2, w_2), \dots, (t_k, w_k)\} \tag{3}$$

where t_i is a term entered by user in his query and w_i is a weight calculated using a formula presented below. When new term occurs in user query that was not used before, the term is added to the user profile set. Term in this context means a word or a few words used by the user to express his information needs.

The adaptation method is the same as in our previous work [8], so here we present only a short overview of the update idea. User profile is updated after a quite big number of user activations, not after each user session. In every session, mean weight

of each term that has occurred in user queries is calculated and relative change of weights values $\Delta w_{t_i}(s)$ between two consecutive sessions is computed. New value of weight is calculated using the following formula (4):

$$w_{t_i}(s+1) = \alpha \cdot w_{t_i}(s) + (1-\alpha) \cdot \left(\frac{A}{1+B \cdot \exp(-\Delta w_{t_i}(s)+C)} \right). \quad (4)$$

where: $w_{t_i}(s+1)$ is a weight of term t_i in user profile in session $s+1$; A , B , C and α are parameters that should be attuned in experimental evaluation. New weight is a linear combination of the old weight and the relative change transformed by sigmoid function. The presented algorithm meets the following properties: if a term is often retrieved by the user, he is interested in it and the weight of this term is increasing. On the other hand, if a term is not occurring any more in user queries, the weight of this term is decreasing.

4 Relevance Judgment Simulation

In most agent-based information retrieval systems, the user profile is updated based on the documents that the user has marked as relevant to him. In our investigations we assume that we do not have any real user that can select useful documents so we need to estimate relevance degree based on the user preferences.

We assume that user preferences *Pref* is a set of 10 weighted terms. The weight represents assumed degree of user interest in selected term. To simplify further calculations, the weights are normalized and all of them meet the condition:

$$\begin{aligned} Pref &= ((t_1, v_1), (t_2, v_2), \dots, (t_{10}, v_{10})) \\ 0,5 &\leq v_1 \leq v_2 \leq \dots \leq v_{10} \leq 1 \end{aligned} \quad (5)$$

As in real case, user preferences are changing with time. In our case weights are randomly selected with the uniform distribution from the interval of [0,5; 1]. We assume that a single change is not greater than 10% of current value and can not be greater than 1 or less than 0,5. From time to time the term with the least weight is replaced by new term and weight is also randomly selected. As Clarkea's research shows [3], in many retrieval systems more than 85% of the queries consisted of three terms or less. Based on this result, in our investigations we assume that the user query is composed of 3 terms. From retrieved documents' list, the relevant are selected and they are used to update the weights in user profile.

The most popular way of document description is its statistical characteristics. A document is treated as a vector in vocabulary vector space model with some numerical measure. It could be a binary approach: if a term occurs in the document, the value of this term is 1, otherwise, it is 0 (e.g. like in Binary Independence Model [7]); other models are using some measures based on TF-IDF measure to calculate the importance of term in selected document.

In our database we do not have access to the content of the book but only descriptions of issues as presented in previous section. Moreover we assume that we do not have a real user that would read this book or even its description and would tell us if this position is relevant to its query and needs or not. For this reason, a novel approach of relevance establishment is proposed.

As described in [9], user submits his query q into the search engine module of agent system. The system retrieves a list of documents. Each document d_i from the list needs to be compared with user information needs expressed by the terms from user preferences (Fig. 1). Based on the weights of terms from user preferences, the probability that document d_i is relevant to the user query q is calculated as follows:

$$P(d_i = rel | q) = \frac{1}{j} \sum_j P(d_i = rel | q_j) = \frac{1}{j} \sum_j \left(\frac{1}{L} \sum_{l=1}^L v_j \cdot P(q_j \in a_l) \right) \tag{6}$$

where:

q_j is the j -th term from user query q

a_l is l -th appellation of considered document; $l \in \{1, L\}$

The interpretation of presented formula 6 is that probability that considered document is relevant to the user query is a sum of probabilities that this document is relevant to each of query terms. Probability that document is relevant to a term can be calculated as a mean value of user interests v_j connected with this term multiplied by probability that the term occurs in the document description. The value of $P(q_j \in a_l)$ is defined as follows:

$$P(q_j \in a_l) = \begin{cases} 1 & \text{if } q_j \subset \{a_m^l, tit^l\} \\ 0.75 & \text{if } (q_j \subset \{a_{wg}^l, a_{wh}^l\} \text{ AND } q_j \not\subset \{a_m^l, tit^l\}) \\ 0.5 & \text{if } q_j \subset a_s^l \text{ AND } q_j \not\subset \{a_m^l, a_{wg}^l, a_{wh}^l, tit^l\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Formula 7 says that the document is more relative to the searched query term if this term occurs in document description and the probability of such situation has a greater value when it occurs in the more important part of document description.

Proposed measure is treated as a probability that presented document is relevant to user query and information needs because it is normalized and its value has easy interpretation: when value of this probability is close to 0, the document is not relevant despite of the fact that specialized search engine recommend this book. When this probability is greater than some assumed threshold, considered issue is accepted as a relevant.

Intuitively the probability that document is relevant to user query should be a product of probabilities that this document is relevant to each term in the query. Unfortunately, in this case the problem could arise because when document contains in its description only one or two terms from user query in appellations, a value of such measure will be 0 that is not expected and not adequate result. Such situation is possible if document with all terms entered by user is not present in the database. For this reason, we have decided to calculate a mean value of these probabilities.

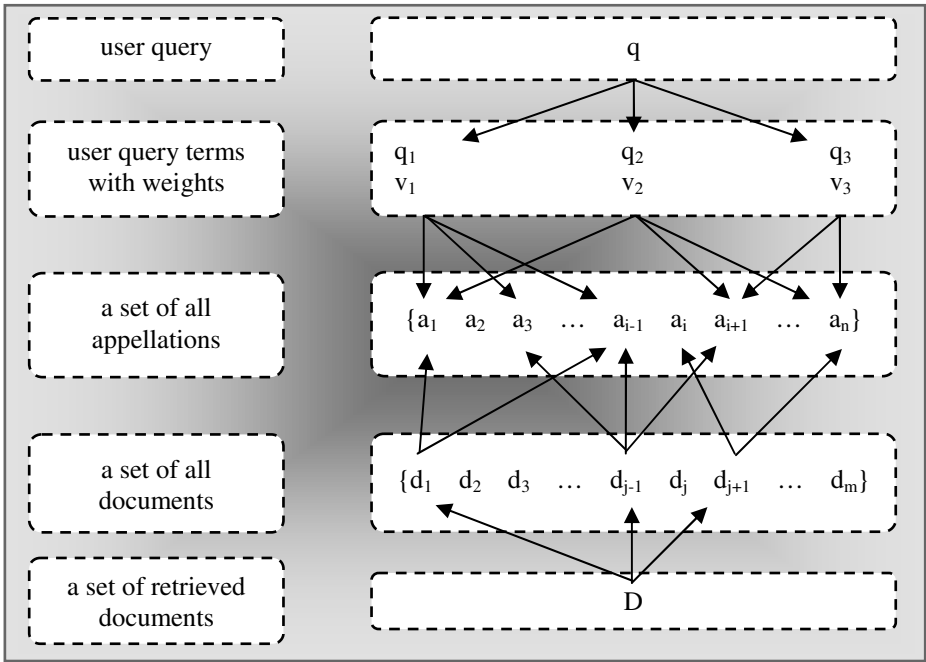


Fig. 1. The model of user query and retrieved documents set with their descriptions. User submits a query q consisting of a few terms and the search engine module of agent system returns a set of documents D . Each document is described by the title (not marked on the figure) and a set of appellations. Probability that the document is relevant to a query takes into account the occurrences of search terms in book’s description.

5 Experimental Evaluation

In our experiment we would like to show that the presented user profile and the method to update are effective in the following terms.

We assume that the user is represented as a list of his preferences with weights that correspond to real degree of user’s interest in particular term. These weights are used to judge relevance of documents retrieved by the system as described in previous section. The agent system is gathering information about relevant documents to build the user profile and update it according to changing user requests. Terms entered by the user are saved in user profile with appropriate weights. We are trying to show that updating user profile with following user sessions decreases the distance between user profile and users’ preferences. In other terms, the built user profile is coming closer to user preferences even if they are changing slightly with time (in this case with consecutive sessions).

The experiment was performed using database of books from the library as described in section 3. The experiment was performed according to the following plan (similarly as in our previous work):

1. Determine user preferences *Pref*.
2. Generate a query *q* – select randomly 3 terms from user preferences set $\{t_1, t_2, \dots, t_{10}\}$, set these terms in proper order according to decreasing weights and send query to search engine.
3. From the return list get document description $d = \{a_1, a_2, \dots, a_L\}$.
4. Calculate the probability that considered document is relevant to user query using formula (5).
5. Mark documents that meet the condition $P(d_i = rel | q) \geq \rho$; $\rho \in (0; 1)$ as ‘relevant’; save relevant documents in the set $D = \{d_1^{(s)}, d_2^{(s)}, \dots, d_n^{(s)}\}$.
6. After 3 blocks of 10 sessions update the profile using adaptation method [8].
7. Calculate the distance between user profile and user preferences using Euclidean distance between those two extended vectors.

Table 1. A sample of user query and results

User query	Description of documents obtained for query
theory knowledge learning	Futures thinking, learning, and leading: applying multiple intelligences to success and innovation, Irving H. Buchen. <i>Cultural pluralism in the work</i> <i>Multiple intelligences</i> <i>Knowledge Management</i> <i>Organization of self-learning</i>
	The theory of organizational changes in the companies of the information age (some aspects and tools), Kazimierz Krupa. <i>Companies - organization</i> <i>Organizational innovations - management</i> <i>Organization of self-learning</i> <i>Knowledge management - Information systems</i>

We have randomly generated the first user preferences:

$Pref(0) = \{(database, 0.81), (management, 0.65), (learning, 0.63), (network, 0.9), (simulation, 0.86), (theory, 0.83), (hybrid, 0.86), (semantic, 0.71), (knowledge, 0.5), (model, 0.53)\}$

Terms for user preferences was obtained from the librarian specification called “Classification of sciences”, available online at Library homepage [16], from the domain of “Automation, Cybernetics and Computer Science”. Weights for terms of user preferences were generated randomly.

User queries were generated by selecting 3 terms from the set of preferences *Pref* in current session. The exemplary user query with retrieved results is presented in Table 1. Each session has contained 5 users’ queries. After each block of 10 sessions, the weights of terms were changing in range of 10% and the term with the lowest weight was replaced by a new term and a new weight for its term was generated. For each document the probability of relevance was calculated and from the set of relevant documents after each 3 blocks of sessions, the user profile was updated.

To check the effectiveness of our profiling method and measure of relevance probability, a series of experiments was performed. We have used extended Euclidean

distance measure between user preferences and user profile. In the adaptation process the following parameters were used: $\alpha=0.4$, $A=1.0$, $B=1.0$ and $C=3.0$. They were tuned during earlier research. In the relevance judgment algorithm we have explored what is the influence of threshold ρ on this distance. The results are gathered in the diagrams on Fig. 2.

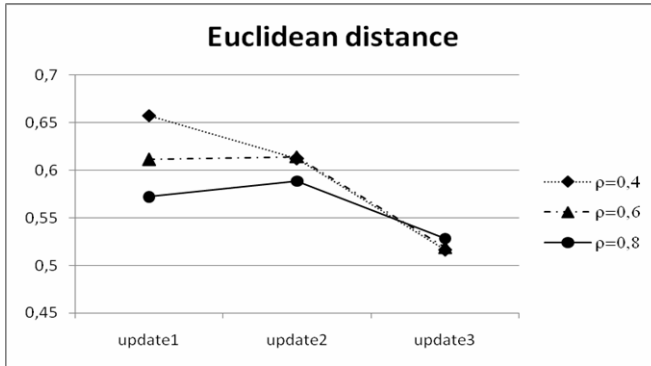


Fig. 2. Euclidean distances between user preferences and user profiles in subsequent updates

Fig. 2. summarizes the Euclidean distance between user preferences and the built user profile updated after 3rd, 6th and 9th block of sessions. The calculated distance is decreasing with subsequent updates, what is a desired result. On the diagram we present three series obtained from different values of threshold ρ . This parameter is responsible for the number of relevant documents: if ρ is smaller, more documents are taken into account in profile updating process and greater ρ means that only the most relevant documents have influence on the updating process. As one can see on the diagram, the best trend is connected with smaller value ρ . When threshold is increasing ($\rho=0.6$ and $\rho=0.8$), the starting points are smaller because only the most relevant documents are important in profile building.

Unfortunately, for these thresholds the distance in second update is almost the same or even a little greater because there can be only a few documents treated as relevant and, in this situation, longer sessions or more blocks of sessions should be taken to a single update process. When we do not start with an empty profile but only update the existing one (difference between 2nd and 3rd update), the distances for all threshold value are decreasing.

In future works more blocks of sessions should be generated to check how relevance algorithm influences on user profile and on the measured distance between user preferences and the built profile. Also some properties of proposed relevance probability and of the update method should be tested.

6 Conclusion Remarks

In this paper we have presented the Personalization Module of Agent-based Personal Assistant and a method of user profile adaptation. The simulation of user behaviour

and an algorithm of relevance judgment based on user preferences are proposed. Performed experiments show that our approach is an effective way of agent-based user profiling in terms of the similarity of user preferences and built profile.

Acknowledgments. This research was partially supported by European Union within European Social Fund under the fellowship no. MK/SN/160/III/2010/U and by Polish Ministry of Science and Higher Education under grant no. N N519 407437.

References

1. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. *Information Processing and Management* 43, 866–886 (2007)
2. Crestani, F., Lalmas, M., Rijsbergen, C.J., Campbell, I.: Is This Document Relevant? ...Probably?: A Survey of Probabilistic Models in Information Retrieval. *ACM Computing Surveys* 30(4) (1998)
3. Clarkea, C.L.A., Cormackb, G., Tudhope, E.A.: Relevance ranking for one to three term queries. *Information Processing & Management* 36, 291–311 (2000)
4. Greisdorf, H.: Relevance thresholds: a multi-stage predictive model of how users evaluate information. *Information Processing and Management* 39, 403–423 (2003)
5. Jung, S., Herlocker, J.L., Webster, J.: Click data as implicit relevance feedback in web search. *Information Processing and Management* 43, 791–807 (2007)
6. Kiewra, M.: The hybrid method for content recommendation in a hypertext environment, Ph.D. dissertation, Wroclaw University of Technology (2007)
7. Lavrenko, V., Croft, W.: B.: Relevance-Based Language Models. In: *SIGIR 2001*. ACM, New Orleans (2001)
8. Mianowska, B., Nguyen, N.T.: A Method for User Profile Adaptation in Document Retrieval. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) *ACIIDS 2011, Part II*. LNCS, vol. 6592, pp. 181–192. Springer, Heidelberg (2011)
9. Mianowska, B., Nguyen, N.T.: A Framework of an Agent-Based Personal Assistant for Internet Users. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6070, pp. 163–172. Springer, Heidelberg (2010)
10. Pan, J., Zhang, B., Wang, S., Wu, G., Wei, D.: Ontology Based User Profiling in Personalized Information Service Agent. In: *CIT 2007*, pp. 1089–1093. IEEE, Los Alamitos (2007)
11. Rijsbergen, C.J.: *Information Retrieval*, University of Glasgow (1979)
12. Robertson, S.E., Maron, M.E., Cooper, W.S.: Probability of Relevance: A Unification of Two Competing Models for Document Retrieval. *Information Technology: Research and Development* 1, 1–21 (1982)
13. Story, R.E.: An Explanation of the Effectiveness of Latent Semantic Indexing by Means of a Bayesian Regression Model. *Information Processing and Management* 32(3), 329–344 (1996)
14. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive Web Search Based on User Profile Constructed without Any Effort from Users. In: *WWW 2004*, New York, USA (2004)
15. Wallis, P., Thom, J.A.: Relevance judgments for assessing recall. *Information Processing and Management* 32(3), 273–286 (1996)
16. Main Library and Scientific Information Centre in Wroclaw University of Technology, <http://www.bg.pwr.wroc.pl/>
17. WordNet Ontology, <http://wordnet.princeton.edu/>

A Multi-agent Organizational Model for Grid Scheduling

Inès Thabet^{1,2}, Issam Bouslimi^{1,2}, Chihab Hanachi¹, and Khaled Ghédira²

¹ Institut de Recherche en Informatique de Toulouse IRIT, UMR 5505,
Université Toulouse 1 Capitole, 2 rue du Doyen-Gabriel-Marty, 31042 Toulouse Cedex 9
{Ines.Thabet, Issam.Bouslimi, Chihab.Hanachi}@irit.fr

² Stratégie d'Optimisation des Informations et de la Connaissance,
Université de Tunis, 92, Avenue 9 avril 1938, Tunis - 1007, Tunis-Tunisie
Khaled.Ghedira@isg.rnu.tn

Abstract. Multi-agent technology provides high level organizational concepts (groups, roles, commitments, interaction protocols) to structure, coordinate and ease the adaptation of distributed systems efficiently. This paper proposes to model a grid scheduling system as a multi-agent system organization. The resulting organizational model, based on the Agent Group Role meta-model of Ferber, is evaluated at the conceptual and implementation level. At the conceptual level, we evaluate the efficiency, robustness and flexibility of our model. At the implementation level, the analysis and the evaluation of our proposition, done through simulations, show its efficiency.

1 Introduction

One motivation of Grid computing [1] is to aggregate the power of widely distributed computational resources and thus to offer the possibility to solve very large problems. To achieve this goal, an efficient Grid Scheduling system is required. The *Grid scheduling system* is in charge of allocating users' applications tasks on a set of feasible and adequate resources following an objective function.

Scheduling systems are difficult to build due to several constraints and notably the dynamicity of the environment and the complexity of the interactions among the different components. These interactions are of paramount importance for finding resources, allocating them tasks and for supporting adaptation in case of disturbances.

The *complexity* of the interactions is due to the fact that the scheduling system is made of several *heterogeneous* and *autonomous* components that have to interact frequently and cooperate at a high level to perform efficiently users' applications in a *disturbed* environment. The *heterogeneity* is due to the different roles involved in a scheduling system (grid scheduler, local scheduler and resource) and the huge number of their occurrences with different features and capacities. Besides, these different components are most of the time *autonomous*, since each one has its own management and access policy. The scheduling may also be *perturbed* by the *dynamicity* of the grid. This dynamicity is due to the open character of the grid (resources can join or leave freely) and to performances variability of the resources.

Giving this context, one of the main issues to face is *how to design, implement and deploy an adaptive grid scheduling system taking into account the previous difficulties*. Moreover, we should deal with this issue at a *macro level* (regulation of the components) since we can't control and act on the internal behavior of each component (micro-level) which are dynamic, huge and above all unknown a priori.

Unfortunately, the examination of the state of the art shows the insufficiency of existing systems to face the previous issue. While agent technology has already been used with success [2] to manage the distribution and the autonomy of the Grid (see the AGEHC system [3], the CONOISE project [4], the AGRD_NFRP system [5]), most of these systems do not take into consideration the dynamicity of the environment. However, we must mention the AppleS agent system [6] that implements an adaptive scheduling. The main drawback of AppleS is that it has to be integrated in the application to be executed and generates a lot of works for the developers. Besides, all these systems impose an internal structure to the agents and do not consider their organization and regulation to reach the system objective.

In order to tackle this issue, we choose to *model the grid scheduling system with an agent organizational perspective* [7] as a means to structure and regulate the scheduling system. This perspective has the advantage of abstracting a system at a high level with macro and social concepts such as roles, interaction protocols, groups, obligations, etc. Not only this organizational model should improve the efficiency of the scheduling system at run time, but it could also be considered as a conceptual tool to capture, model and implement the structure and the behavior of this system in a modular way. Moreover, the concept of agent organization is now mature, and techniques to make it flexible and adaptive are today available [8] and could be used with benefit for the adaptation of the grid scheduling.

More precisely, the proposed organizational model follows the Agent Group Role (AGR) meta-model of Ferber [9]. This model has been designed, implemented and evaluated. Our contributions are threefold:

- *An organizational model* for grid scheduling, which identifies the roles involved in our system, the links that exist between them and the protocols that rule their interactions. This model is general enough to capture any grid scheduling system.
- *Performance criteria* for evaluating our proposed organizational model based on the adaptation of [10] and [11] works. The criteria include structural and statistical aspects. The *structural aspect* is based on the communication network topology. The *statistical aspect* is concerned with the communication and execution time.
- *An implementation and evaluation* of our model. The implementation is based on the Madkit platform [12] that integrates the AGR meta-model. Our evaluation measures the performance criteria mentioned above.

The remainder of this paper is organized as follows. In Section 2, an overview of the proposed organizational model for grid scheduling is presented. We propose performance criteria and the evaluation of our organizational model for grid scheduling in Section 3. Finally, we summarize and lay out the future work.

2 Overview of the Organizational Model for Grid Scheduling

In order to specify and describe the functioning of the grid scheduling system (GS) and more precisely how the agents interact, we choose to model the GS with an organizational view. Introducing organizational abstractions makes it possible to structure and regulate agent's interactions in order to organize their works, limit conflicts and reduce communication cost. For that purpose, we have used the Agent Group Role (AGR) meta model which is appropriate to the Grid context. In AGR, *agents* are actives communicating entities that can participate in several *groups* (communities) in parallel and can play one or several *roles* in each group. No constraint is placed on the internal structure of the agent. AGR has several advantages useful in our context: it eases the *modularity* through the organization of agents' tasks and their interactions in logical entities. Also, AGR allows the design of *open* systems such as grid since new agents could enter or leave the system provided that they could play predefined roles. The system can also be designed with *heterogeneous* components since agent internal architectures are not constraints. Finally, the *reusability* is facilitated since the organizational approach identifies and makes available recurring interaction patterns which become reusable.

The remainder of this section first presents the Grid Scheduling architecture and the proposed organizational model for Grid Scheduling.

2.1 General Architecture of a Grid Scheduling System

Following previous work, we defined a multi-agent based architecture for the grid scheduling system. See [13] for details. In this section we describe the architecture.

The *User Interface* allows the interaction between the user and the Global Scheduler (GS) Agents. Using the user interface, the user submits a set of applications made by a set of coordinated tasks, to be executed on the Grid, to the GS.

The *Global Scheduler Agents* (GS) are responsible for managing one or several domains, scheduling application's users on a set of adequate resources. A GS orchestrates applications following their processes and selects the best pool of resources in order to optimize its objective function. Moreover, the GS is responsible for monitoring, if necessary adapting the grid functioning in case of disturbances, collecting the execution results and reporting them to the user. The adaptation mechanism will be detailed in the next section and are basically based on the use of organizational structure adaptation, on rescheduling and on interaction protocols.

The *Local Scheduler Agents* (LS) are responsible for managing one or several resources belonging to the same domain. A Local Scheduler is also responsible for tasks scheduling in a domain according to the local policy. Moreover, the LS is in charge of detecting failure and for adapting tasks' allocation if possible.

Resources provide high computing capabilities to enable task execution. Each resource is managed by a *Resource Agent* (RA) in charge of reporting information (resource state and availabilities, tasks execution progress, execution results) to the Local Scheduler.

2.2 An AGR Organizational Model for Grid Scheduling

In this section, we describe our organizational model which identifies the actors implied in our system, the links that exist between them, and the protocols ruling their interactions. Our model is composed of the following components (see Fig. 1):

- Five types of groups represented by an ellipse and described below.
- Three types of agents represented by a candle that are: *Global Scheduler*, *Local Scheduler* and *Resource Agents*. The multiplicity of the agents participating to a group is represented by a star inside the candle.
- Five types of roles since the Local Scheduler and the Resource Agents play two different roles in different groups. The defined roles are: *Global Scheduler (GS)*, *Local Scheduler*, *Resource*, *Resource Application* and *Virtual Organization Scheduler* Roles. Role is represented as a hexagon and a line links this hexagon to agents. The Resource Application Role concerns a dedicated resource that offers one or several services (here the service is the task execution) with a certain quality (dependent of the resource execution capacity). The mapping between the application’s tasks types (skill or capacity required by the resource to execute this task) and the resource capacities entails a virtual organization formation made by specialized and distributed physical resources. When a Virtual Organization is formed, an agent playing the role Virtual organization Scheduler is launched by the GS in order to orchestrate the tasks in the virtual organization.

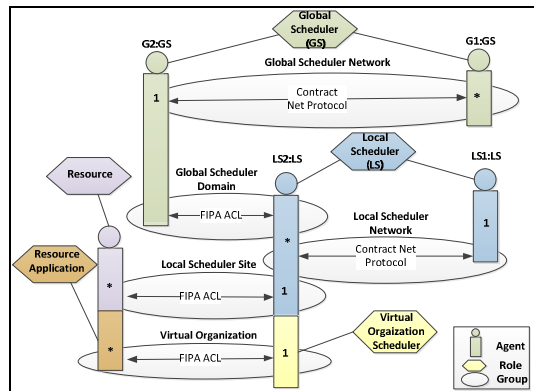


Fig. 1. Organizational Multi-agent based Grid Architecture

The communication between agents in different groups follows the FIPA-ACL language that supports high level communication between the Grid components.

Let us introduce the defined groups and detail how does each group operates.

The **Global Scheduler (GS) Network** allows the GS to cooperate in order to allocate application’s tasks efficiently when resources, under a GS control, are not available or in the case of disturbance. In this context, GS agents cooperate in order to sub-contract tasks’ executions using the *Contract Net (CN) Protocol* [14]. The use of the CN is justified since it is one of the most flexible and efficient negotiation mechanisms and it eases tasks distribution. Moreover, it allows the GS managing the

contract to choose the better GS bidder regarding its objective. In our current implementation, the GS optimizes the application makespan.

The **Global Scheduler (GS) Domain** Group is composed of a GS and its Local Scheduler (LS) agents. This group allows the GS to allocate the application's tasks to the adequate LS, to be executed on its resources, based on the GS objective, resources states and task requirements. Moreover, this group allows the LS to send execution results and resources states, and to inform its GS if problems occur on resources.

The **Local Scheduler (LS) Site** Group is composed of the Local Scheduler and the resources at its disposal (in the local site). This group allows a LS to allocate the submitted tasks to the local resources, to collect the resources states (busy, free, broken), the execution progress and results, etc. In the case of resources breakdown, the LS try to reschedule the failed tasks on other resources belonging to its group according to the local resources state. If there is no adequate resource, the LS uses the Local Scheduler Network to find adequate LS acquaintance for tasks execution.

The **Local Scheduler (LS) Network** Group allows the LS agents to cooperate efficiently in order to sub-contract tasks execution when resources, at the LS disposal, are not available, busy, broken, etc. The LS agents cooperate using the Contract Net Protocol. The LS initiator evaluates the received offers (execution estimation) and sends tasks to the selected LS or notifies the Global Scheduler about the failure. In this last case, the GS performs an adequate adaptive action.

Let us notice that in the previous groups, adaptation is performed by to the use of high-level protocols, like contract net protocol that eases task allocation even if the Grid environment evolves. The use of interaction protocols allows selecting dynamically resources to face the failure without varying the organization's structure.

The **Virtual Organization (VO)** Group is composed of specialized resources, dedicated to a type of task, and by a Local Scheduler playing the role of Virtual Organization Scheduler (VS). The VS is responsible of coordinating the functioning of the Resources Application (dedicated resource) and monitoring their execution. The Virtual Organization formation made by the Global Scheduler depends on the offered services, QoS and resources availabilities. The adaptation in this case consists in reorganizing the virtual organization by removing overloaded resources, adding new better specialized resources, replacing dedicated resources by others, etc. The adaptation results in the modification of the structure of the virtual organization.

3 Evaluation of our Model

3.1 Performance Metrics

In this section we introduce the performance criteria used in our approach and the evaluation of our model for grid scheduling based on the following criteria. These criteria are organized according to two aspects: the *structural* and the *statistical* level:

The *structural level* concerns the communication topology of the organization and is based on the Grossi's measures [10] that will be described below. These measures allow to proof organizational qualities such as the *flexibility* (the capacity to adapt easily to changing circumstances), the *robustness* (how stable the organization is in the case of anticipated risks) and the *efficiency* (the amount of resources used to

perform its tasks) by using relations between roles. In our case, we will evaluate these qualities and prove that our model is sufficiently robust and efficient.

The *statistical level* concerns the performance's evaluation of our organizational model at run time. To this end, two metrics will be observed: the *application execution time* and the *number of messages* in scenarios without perturbations and with perturbations. The comparison of the number of messages with and without perturbations is commonly used for evaluating adaptive multi-agents system [11].

Lets' us details the *Structural level* based on Grossi's [10] measures. Evaluating the organizational structure qualities involves three steps:

1. The first step consists in building a role graph of the organization based on the possible relations between two roles. To do that, Grossi introduces three dimensions characterizing these relations:
 - The *power structure* defines the task delegation patterns. The existence of power link between agents "a" and agent "b" means that every delegation of tasks from "a" to "b" ends in creating an obligation to "b".
 - The *coordination structure* concerns the flow of information in the organization.
 - The *control structure*. An agent "a" controls an agent "b" if "a" monitors "b" s activity and "a" possibly takes over the tasks which "b" has not accomplished.
2. Secondly, for *each dimension* (power, coordination, control), a set of concepts and equations from the graph theory are applied in order to measure specific qualities (robustness, flexibility and efficiency) of organizational structures (OS):
 - The *connectedness* of an OS shows the connection degree between roles. The more this degree is high, the more the structure can be divided in logical groups.
 - The *economy* of an OS expresses how to keep the structure connected while minimizing the number of links between roles (redundant links must be avoided).
 - The *univocity* allows us to have an idea on the degree of ambiguity in the OS. For example stating that an agent "a" controls an agent "b" and in the same way, "b" controls "a" generates some ambiguity.
3. Finally, these previous concepts are compared with optimum values defined by Grossi (See [10] for more details) in order to measure the quality of the OS.

3.2 Evaluation of Our Organizational Model

In this section we describe the structural and statistical evaluation of our model.

3.2.1 Structural Evaluation

According to the dimensions described above (power, coordination, control), we generate the role graph (Fig. 2) corresponding to our organizational model. Our graph includes the roles described in the section 2.2. Also, we distinguish between the role "Global Scheduler (GS) Initiator" that is a GS playing the role of tasks manager and the role "Global Scheduler Contractor" corresponding to bidders. The same separation has been made for the Local Scheduler (LS) role.

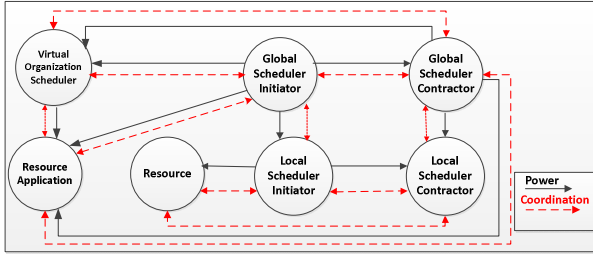


Fig. 2. Role graph

Our graph includes only the power and coordination dimensions represented as edges:

- *Power* expresses the delegation mechanism. In our context, the GS Initiator can delegate tasks to all the other roles except to the resources only managed by LSs (Initiator and Contractor). The Virtual Organization Scheduler (VS) has a power relation with the Resources Application. The LS Initiator also has a power relation with its acquaintances (LS contractor) and with the Resources under its control.
- *Coordination* is a symmetric relation in our graph. It consists in information exchange or protocols-based interaction between roles. It concerns the same couple of roles as the power relation.

The *Control* relation is not present in our organization, since we do not model the evaluation of agents' behavior by their acquaintances. In future work, we intend to refine our model to include this dimension. This assumes to model the continuous monitoring of the agent work to check whether agents are doing their tasks as agreed.

Following this role graph and the set of equations introduced in [10], we compute the three following organizational qualities: the robustness, flexibility and efficiency as shown in the tables below:

Table 1. Structural measures of robustness

Optimum Values	Economy _{Coord}	0	Univocity _{Power}	0	Connect _{Coord}	1
Obtained Values		5/9		1/7		1

Table 2. Structural measures of flexibility

Optimum Values	Connect _{Power}	0	Connect _{Coord}	1	Economy _{Coord}	0
Obtained Values		1		1		5/9

Table 3. Structural measures of efficiency

Optimum Values	Economy _{Power}	1	Economy _{Control}	1	Economy _{Coord}	1
Obtained Values		8/9		1		5/9

Comparing our results to the optimum ones, we can state that our system is sufficiently *robust* and *efficient* but not enough *flexible*. Lets' us explain these results:

Robustness of an organization requires a coordination structure highly connected and weakly economic and a power structure weakly univocal (see Table 1). In our organization, roles are well connected ($\text{Connectedness}_{\text{Coord}}$ value equal to the optimum) while avoiding ambiguities ($\text{Univocity}_{\text{Power}}$ value approaches the optimum). An optimal robustness would require a complete connectivity between all nodes. Due to the privacy and local management policy, this complete connectivity is not possible in a grid context. For example each resource in a domain has its own management policy and doesn't authorize the Global Scheduler to interact directly with it.

Flexibility of an organization is related to the ability to easily adapt. So, in order to enhance the flexibility, a low degree of connectedness for the power and a high degree for the coordination dimensions are needed. Considering the connectivity aspect, while the power relation restrains the flexibility (it imposes constraints on the interactions), the coordination one eases the interactions: we reach a trade-off (see Table 2). Considering the economy aspect, we have an average value which corresponds to a moderate flexibility. To obtain a lower value of the $\text{Economy}_{\text{Coord}}$, for more flexibility, we should increase the redundancy of the coordination which is incompatible with the privacy constraint of the grid context.

Efficiency of an organization can be obtained if it is economic (value equal to 1) in all the dimensions. In our case, the roles are well connected while avoiding redundant ones and we approach the optimal efficiency. The $\text{Economy}_{\text{Control}}$ is equal to the optimum one since we don't take into consideration this dimension.

It is important to notice that the framework proposed by Grossi [10], used for analyzing organizational structures (OS), does not take into consideration the number of roles' instances and the links among them. However, in our context, introducing the roles' cardinalities could be more significant and realistic. For example, a broken resource doesn't imply that the resource role dysfunctions, but means only that this specific instance is out of work and therefore, it could be replaced by one another.

Also, obviously, designing an OS that maximizes simultaneously the three qualities cannot be reached. For example, the more the OS' nodes are well and directly connected, the more the OS may be robust and flexible and the less it is efficient.

Since our organization is devoted to the grid scheduling system, we believe that the qualities obtained (sufficiently robust and efficient) are, in our opinion, the most useful for this context.

As our context is highly dynamic and characterized by a frequent resources performances fluctuations and crashes, we are also interested in the capacity of our organization to be *adaptive* (ability of the system to react against the environment changes). The adaptativity cannot be measured using the previous framework. In the next section, we will measure the quality of our model experimentally by simulations and more precisely its *adaptability*. For this purpose, we will compare the nominal functioning of our system and its functioning with disturbances.

3.2.2 Statistical Evaluation

In this section, we describe our experiments and results.

We have implemented a Grid Scheduling Simulator based on the AGR meta-model using Java and the Madkit [12] platform. Our simulator offers the following services:

- It allows users to enter parameters for the simulations (resources with their numbers, capacities, organization in domains, resource’s failure probability, applications’ tasks and their coordination process, submission time, etc).
- It simulates the applications’ tasks scheduling and their executions step by step,
- It finally outputs the statistical data of the performance metrics on a log file.

Our experimental system is configured with twelve agents, illustrated by the figure 3. The Global Schedulers are represented with a square (GS1, GS2), the Local Schedulers are represented with an ellipse and resources with a rounded rectangle (including the CPU speed). The following grid specification is simulated and organized according to our proposed AGR model described in section 2.2.

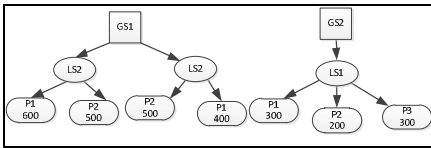


Fig. 3. Case study: agent’s organization

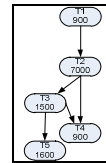


Fig. 4. Case study: application description

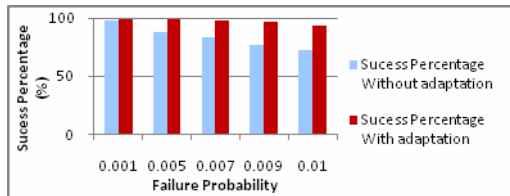


Fig. 5. Execution success percentage

The goal of these simulations is to demonstrate the feasibility (figures 5) and the efficiency (figure 6, 7, 8 and 9) of our model and more precisely of its adaptation.

To demonstrate *the feasibility of our proposition*, we compute the number of successful executions of applications in the case of adaptation comparing to the case with no adaptation. To do that, we consider four same applications (figure 4 shows the application description) and we vary the resources’ failure probability. We measure the percent of successful runs. The total number of run is fixed to 100.

The results show the efficiency of our adaptation model since we obtain better percentage of successful executions with adaptation. We can however notice that we do not reach 100% of success since even with adaptation we can meet situations where some tasks couldn’t be executed by lack of specific resources.

To demonstrate *the efficiency of the adaptation* of our model, we make two experiments. In the first one, we consider only one application and we vary the resource’s failure probability. Figure 6 compares the application execution time with and without disturbances. The results show that the adaptation phase is not costly (at the maximum 27% of the application execution time). Figure 7 shows how the number of communication’ messages increases as the failure probability raises. This evolution is due to the fact that adaptation is mainly based on interactions.

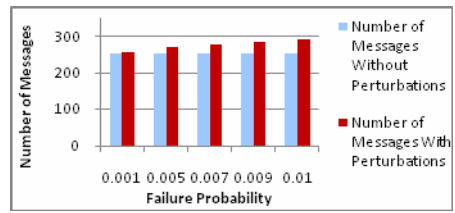
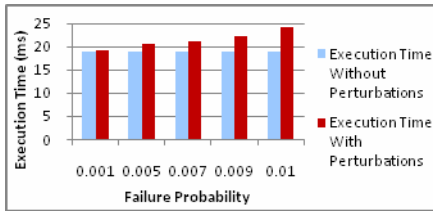


Fig. 6. Comparison of application execution time **Fig. 7.** Comparison of number of messages

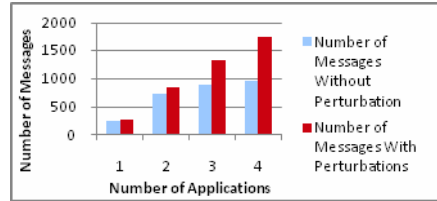
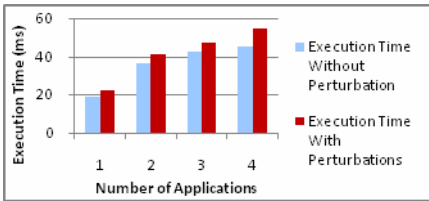


Fig. 8. Comparison of application execution time **Fig. 9.** Comparison of number of Messages

The second experiment varies the number of concurrent applications executed by the system with a fixed resource's failure probability (0.01). Figure 8 shows again that the adaptation time remains low (20% of the execution time) even if the number of applications increases. Figure 9 shows that the number of messages increases with the number of applications in both cases. In case of perturbations, we notice a higher but reasonable number of communications due to the adaptation phase.

4 Conclusion

We have proposed an organizational model to describe and implement a grid scheduling system. This system has been validated: it has been implemented and its design and performances have been discussed. Following the Grossi's framework we have evaluated its conceptual aspect and shown that it is sufficiently robust and efficient. According to our implementation, we have shown the feasibility and efficiency of our approach and more precisely measured the adaptability of our organizational model. In addition, the proposed system presents the following qualitative advantages:

Easy design and implementation of the grid scheduling system. Conventional design of such a system is known to be a difficult task due to the constraints of the environment. The organizational perspective, followed in this paper, constitutes a design support and makes it possible to structure the overall functioning through the attribution of roles and interaction rules to which the agents must conform.

Openness. The use of an organizational perspective eases openness: agents playing predefined roles can enter or leave the grid freely.

Reusable framework. Our work is reusable and could be considered as a first step towards a simulation framework for designing and testing grid organizations. Indeed, designers could use our system to model their own organizations, measure their qualities

(robustness, flexibility, efficiency), tune their organizations in order to fit with their requirements, validate these qualities and finally simulate their functioning.

This work opens three main issues for future research. The first issue we are currently investigating is how to add the *control dimension* and what are the effects on the qualities of our organization. The second issue is related to the *structural evaluation* of our organization. We do believe that a refinement of the theoretical evaluation is needed. Indeed, taking into account the roles cardinalities and the relation's occurrences will certainly influence the evaluation of our model. Finally, our simulations were only limited to four concurrent applications. To measure the *scalability* of our approach, we need to consider more applications and possibly move from a simulation framework to a real grid environment.

References

1. Kesselman, C., Foster, I.: *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, San Francisco (2004)
2. Foster, I., Kesselman, C., Jennings, N.: Brain Meets Brawn: Why Grid and Agents Need Each Other. In: 3th International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 8–15. IEEE Computer Society, Los Alamitos (2004)
3. Shi, Z., Huang, H., Luo, J., Lin, F., Zhang, H.: Agent-based grid computing. *Applied Mathematical Modelling* 30, 629–640 (2006)
4. Patel, J., Teacy, W.T., Jennings, N.R., Luck, M., Chalmers, S., Oren, N., Norman, T.J., Preece, A., Gray, P.M., Shercliff, G., et al.: Agent-based virtual organisations for the grid. *Multiagent and Grid Systems* 1, 237–249 (2005)
5. Muthuchelvi, P., Anandha Mala, G.S., Ramachandran, V.: Agent Based Grid Resource Discovery with Negotiated Alternate Solution and Non-Functional Requirement Preferences. *Journal of Computer Science* 5, 191–198 (2009)
6. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, S.G.J.: al.: Adaptive computing on the grid using AppLeS. *IEEE Transactions on Parallel and Distributed Systems* 14, 369–382 (2003)
7. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J.J. (eds.) *AOSE 2003*. LNCS, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)
8. Dignum, V.: *The Role of Organization in Agent Systems*. Multi-agent Systems: Semantics and Dynamics of Organizational Models. IGI (2009)
9. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: *The International Conference on Multi Agent Systems*, pp. 128–135 (2002)
10. Grossi, D., Dignum, F., Dignum, V., Dastani, M., Royakkers, L.: Structural evaluation of agent organizations. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1110–1112. ACM, New York (2006)
11. Kaddoum, E., Gleizes, M.P., Georgé, J.P., Picard, G.: Characterizing and evaluating problem solving self-* systems. In: *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, pp. 137–145 (2009)
12. Madkit Website, <http://www.madkit.org/>
13. Thabet, I., Hanachi, C., Ghédira, K.: Towards an adaptive grid scheduling: Architecture and protocols specification. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009*. LNCS, vol. 5559, pp. 599–608. Springer, Heidelberg (2009)
14. Smith, R.G.: The contract net protocol. *IEEE Transactions on High-level communication and control in a distributed problem solver* 100, 1104–1113 (2006)

Extending the JADE Agent Behaviour Model with *JBehaviourTrees Framework*

Iva Bojic¹, Tomislav Lipic², Mario Kusek¹, and Gordan Jezic¹

¹ University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Zagreb, Croatia

{iva.bojic,mario.kusek,gordan.jezic}@fer.hr

² Rudjer Boskovic Institute, Centre for Informatics and Computing
Bijenicka 54, HR-10000, Zagreb, Croatia
tlipic@irb.hr

Abstract. Creating modular behaviours in JADE using a traditional method such as the FSM (Finite State Machine) can be a difficult task to achieve. The first issue with FSMs is that they do not allow reusability of logic in different contexts. Secondly, the FSMs do not lend themselves well to concurrency within the execution thread and thus eliminating the possibility for parallel behaviours. Lastly, as the number of states in FSMs becomes increasing large, it becomes difficult to manage them. In this paper we introduce our *JBehaviourTrees Framework* that extends JADE *Behaviours* with BTs (Behaviour Trees) model. BTs are built via the composition of basic tasks increasing the possibility for modularity and code reuse. The proposed approach is verified through a case study concerning a FIPA-Request Interaction Protocol.

Keywords: Behaviour trees, JADE, Finite State Machines, FIPA-Request Interaction Protocol.

1 Introduction

In general, each software agent must be autonomous, proactive, reactive and posses some social skills [14]. Specifically, in this paper we are dealing with software agents implemented in JADE (Java Agent DEvelopment Framework), since JADE implements all basic FIPA (Foundation for Intelligent Physical Agents) specifications. In JADE, agents have full control over their internal states and behaviours and the only way for them to communicate with other agents is by sending ACL (Agent Communication Language) messages. Therefore, they have their autonomy, but also by communicating with others, are able to be social. Moreover, agents' proactiveness can be achieved implementing their behaviours as goal-driven [2]. Although, the design choice of JADE was to keep the agent abstraction simple, without requiring explicit representation of goals and mental attitudes, JADE user community provided different solutions [12,8]. Finally, reactive agents typically are more action oriented as they map their perceptions into actions.

There are numerous methods for building simple, reactive agents based on FSMs (Finite State Machines) [7], HFSMs (Hierarchical Finite State Machines) [5,4] and BTs [10,1] (Behaviour Trees). FSMs are widely used as the main technology when creating the Artificial Intelligence (e.g. in games for nonplayers characters) because of their efficiency, simplicity and expressibility. However, FSMs do not allow reusability of logic in different contexts. Therefore, in order to reuse them, states must be duplicated (causing redundancy) or a great number of complex transitions must be added. Moreover, FSMs suffer from the problem of becoming unmanageable past a certain size, as maintaining the $O(n^2)$ transitions between states becomes an increasingly difficult problem [6].

HFSMs reduce this issue by allowing grouping together a set of states (i.e. super-states) that have common transitions. Different super-states can be then grouped together creating a hierarchy. However, they do not allow reusing of states in different situations because transitions are hard-coded in them. The solution is to use the BTs, which have the same power as HFSMs, but move transitions to external states, so states become self-contained. Every state therefore encapsulates some piece of logic transparently and independently of the context [13].

The main contributions of this paper are summarized below:

- introduction and implementation of BT model into JADE within *JBehaviourTrees Framework*, and;
- its functional evaluation on FIPA-Request IP (Interaction Protocol).







The rest of the paper is organized as follows. Section 2 introduces BTs, while Section 3 presents different types of agents behaviours in JADE and the way how they are used to implement BTs explaining our *JBehaviourTrees Framework* (letter *J* is abbreviation for JADE). Section 4 illustrates FIPA-Request Interaction Protocol case study comparing protocol's standard implementation as FSMs with our implementation of this protocol using *JBehaviourTrees*. Finally, Section 5 concludes the paper and gives an outline for future work.

2 Behaviour Trees

The main building block of BT is a task, instead of state in FSMs and HFSMs. Table 1 shows types of tasks in BTs: *Composite tasks* (i.e. *Selectors*, *Sequences* and *Parallels*), *Decorator task* and *Leaf tasks* (i.e. *Actions* and *Conditions*) [11]. Leaf task performs an *action* or determines if a *condition* has been met. Composite tasks and Decorator are composed of children tasks that can be either composite tasks (i.e. Composites and Decorator) or Leaves.

Execution process in BTs can be described through handling the return status of tasks and their (un)blocking policies (see Table 1). Each task returns a status code indicating success or failure (possibly with error statuses). Leaf tasks return status codes indicating whether they succeeded or failed (e.g. Condition task returns success if condition *C* is met, and failure otherwise), while status code from Composite tasks depends on their children. Moreover, in order to avoid

Table 1. Types of BTs' tasks

Task	Symbol	Execution process	
Selector		succeeds if one child succeeds	(un)blocks when its currently active child (un)blocks
Sequence		fails if one child fails	
Parallel		fails if F children fail (Failure policy); succeeds if S children succeed (Success policy)	blocks when all its children block; unblocks as soon as any of its children unblocks
Decorator		manipulates with the return status of its child	(un)blocks itself and its children
Action		fails if it cannot complete the action	(un)blocks itself
Condition		succeeds if condition C is met	

waiting on some tasks (e.g. while they wait on messages), every single task can be blocked. The *block()* method puts the task in a queue of blocked tasks, while *unblock()* method restarts a blocked task.

2.1 Composite Tasks

The root of BT is basically top-level task that can be then decomposed recursively into sub-tasks accomplished with simpler tasks called *Composite tasks*. Composite tasks are tasks with one or more children, which act as decision nodes within the tree determining which child task to execute. Their execution depends on the execution of their children. When child task is completed and has returned its status code, the Composite decides whether to continue through its other children or whether to stop and return a value.

Selector. Selectors are used to choose the first child that is successful. A Selector will return immediately with a success status code when one of its children ends successfully. Each Selector can have its own criteria for selecting the child task:

- *Probability selector* - chooses a child task based on a probability (random or specified by user), and;
- *Priority selector* - chooses a child task based on the order of children (first child is most important).

Sequence. Sequence sequentially executes all its children in order. It will return immediately with a failure status code when one of its children fails. As long as its children are succeeding, it will keep going. If it runs out of children, it will return success.

Parallel. Parallel task supports execution of concurrent tasks, running all of its children at the same time. A Parallel task has a parallel policy: *Failure policy* or *Success policy* (see Table III). Symbol for Parallel task has letter F indicating how many children have to fail, so Parallel task would fail. Analogously, letter S denotes the number of children that have to succeed in order for Parallel task to be successful.

2.2 Decorator Task

The name *decorator* is taken from object-oriented software engineering. The decorator pattern refers to a class that wraps another class, modifying its behaviour. Decorator in BTs is a type of task that has one single child task and modifies its behaviour. It takes the original task and adds decorations, i.e. new features. For instance, Decorator can be used to: limit the number of times a task can be run, prevent task from firing too often with a timer or restrict the number of tasks running simultaneously.

2.3 Leaf Tasks

Leaf tasks are tasks with no children: *Actions* and *Conditions*. An Action task performs an action, while Condition task determines if a condition C has been met. By separating Leaf tasks into Actions and Conditions, BTs become more easily understood and adaptable, extending behaviour modularity and functionality. Therefore, Composite tasks can be used in a way that allows them to act as pre-conditions or assumptions for other tasks.

3 *JBehaviourTrees Framework: Implementing BTs into JADE*

In JADE, *Behaviour* class is an abstract class for modelling agent behaviours. It is responsible for the basic behaviour scheduling and state transitions, such as starting, running, blocking, restarting and terminating JADE's behaviours. Since *Behaviour* class, along with its subclasses (e.g. *CompositeBehaviour*, *SimpleBehaviour* and *WrapperBehaviour*) provide whole functionality and control of behaviours execution process, we explain how we used that mechanisms in our *JBehaviourTrees Framework*.

3.1 Implementing BTs Tasks Functionality

Our *JBehaviourTrees Framework* consist of six classes: *SelectorTask*, *SequenceTask*, *ParallelTask*, *DecoratorTask*, *ActionTask* and *ConditionTask* classes. We used *CompositeBehaviour* class to implement set of Composite tasks, *WrapperBehaviour* class is extended in classes that implement functionality of *DecoratorTask*, and finally, Leaf tasks are implemented as *SimpleBehaviours*.

Composite tasks. *SelectorTask* and *SequenceTask* extend *SerialBehaviour*, a composite behaviour with serial children scheduling. They both implement *checkTermination()* method that checks whether to terminate. *SelectorTask* terminates when one child succeeds (returning success) or ends when all children are executed and none of them succeed (returning failure), while *SequenceTask* terminates when one child fails (returning failure) or ends when all children are executed and none of them failed (returning success). *ParallelTask* is implemented extending *ParallelBehaviour* class, a composite behaviour with concurrent children scheduling. *ParallelTask* terminates (and returns success) when a particular condition on its sub-behaviours is met (i.e. when all children are done, N children are done or one child is done). Otherwise it ends returning failure.

Decorator task. *DecoratorTask* is implemented extending *WrapperBehaviour* class. This class allows modifying on the fly the way an existing behaviour object works, and therefore provides a good decorator design pattern for BTs Decorator task implementation.

Leaf tasks. Both, *ActionTask* and *ConditionTask* are implemented as *OneShotBehaviours*, that completes immediately after its *action()* method is executed exactly one time. When some condition must be checked repeatably, or some action executed more than once, *DecoratorTask* can be used to determinate the number of times leaf task must be run.

3.2 Controlling BTs Tasks Execution Process

In BTs, the route from the top level to each leaf represents one course of action, and the execution of BT evaluates those courses of action in a left-to-right manner performing a depth-first search. The control of this execution process can be achieved using JADE's behaviours scheduling, control handling and (un)blocking mechanisms.

Scheduling. If an agent in JADE, has more independent behaviours (added using *addBehaviour()* method in base *Agent* class), their execution is controlled by JADE's scheduler. Scheduler, implemented by the base *Agent* class, maintains a set of active behaviours that are then executed in a round-robin manner. Behaviours, that are not independent behaviours, but sub-behaviours of *CompositeBehaviour* (e.g. like tasks in BTs), have their own scheduling policies realized through "control handling mechanisms".

Control handling mechanisms. Composite and Decorator tasks in BTs are used to control the flow within the tree, while Leaf tasks execute code returning success or failure. In all cases, except when the task is currently running, control is passed back to the parent task. The parent task then handles the given return status of its child passing it up to its parent.

In JADE, the *CompositeBehaviour* class provides only a common interface for children scheduling, but does not define any scheduling policy. This means that the actual operations performed by executing this behaviour are not defined in the behaviour itself, but inside its children. Scheduling policies, defined by its subclasses (*SerialBehaviour* and *ParallelBehaviour* classes) are already explained in Section 3.1.

As for the returning codes, the *Behaviour* class provides a place-holder method named *onEnd()*. This method must be overridden returning an *int value* that represents a termination value for that behaviour. This int value denotes either that behaviour has completed its task successfully, or that it was terminated after some error had happened.

(Un)blocking mechanisms. In order to maintain the list of active tasks ready for execution, each task must be classified as *running*, *blocked* or *terminated*. Task is in runnable phase until it is blocked or terminated. When blocked, it enters in inactive mode and is not in list of active tasks.

In JADE, a behaviour is active when executing its *action()* method. By using *block()* method, behaviour can be put in a queue of blocked behaviours, and can be unblocked when *restart()* method is called, while *handle()* method handles block/restart events in behaviours through *RunnableChangedEvent* class. *RunnableChangedEvent* class is used to notify interested behaviours when certain behaviour changes its runnable state sending this information to behaviour's parent (upward notification) or to behaviour's children (downward notification). Finally, behaviour is terminated when its *done()* method returns *true*.

3.3 Achieving BTs Task Collaboration

Agents in JADE can communicate transparently sending ACL messages. Their communication is based on an asynchronous message passing paradigm. But, for inner-agent communication (i.e. collaboration), between behaviours within one composite behaviour (i.e. *CompositeBehaviour* class), JADE provides private data store for each behaviour through *DataStore* class. In this way a blackboard architecture for *CompositeBehaviours* and their children is realized and it is ground for data sharing among our JADE BTs tasks.

The blackboard is a useful external data store for exchanging data between behaviours (i.e. tasks). When agents communicate by writing and reading from the blackboard, they are independent from each other, and that is a better way than calling methods. Moreover, having all behaviours communicate in this way, allows a usage of existing data in novel ways, making it quicker to extend the functionality of implementation.

4 Case Study: FIPA-Request Interaction Protocol

FIPA-ACL [3] specifies a collection of communicative acts where semantics of each act is specified in terms of a set of feasibility preconditions and a set of

REs (Rational Effects). Interaction protocols specify agent communication by defining sequences of speech acts that can be used for specific type of interactions between agents. In this section we explain the FIPA-Request IP.

The FIPA-Request IP [3] allows one agent (i.e. *Initiator*) to request another (i.e. *Participant*) to perform some action. The Participant processes the request and makes a decision whether to accept it or refuse it. Once the request has been agreed upon, then the Participant must communicate either: a *failure*, an *inform-done* or an *inform-result*.

4.1 FIPA-Request IP Implemented as FSM

In JADE, FIPA-Request IP is implemented with two classes: *AchieveREInitiator* and *AchieveREResponder* (see Figure 1). *AchieveREInitiator* class extends *Initiator* class that is implemented as *FSMBehaviour* (implements FSM model into JADE), while *AchieveREResponder* directly extends *FSMBehaviour* class.

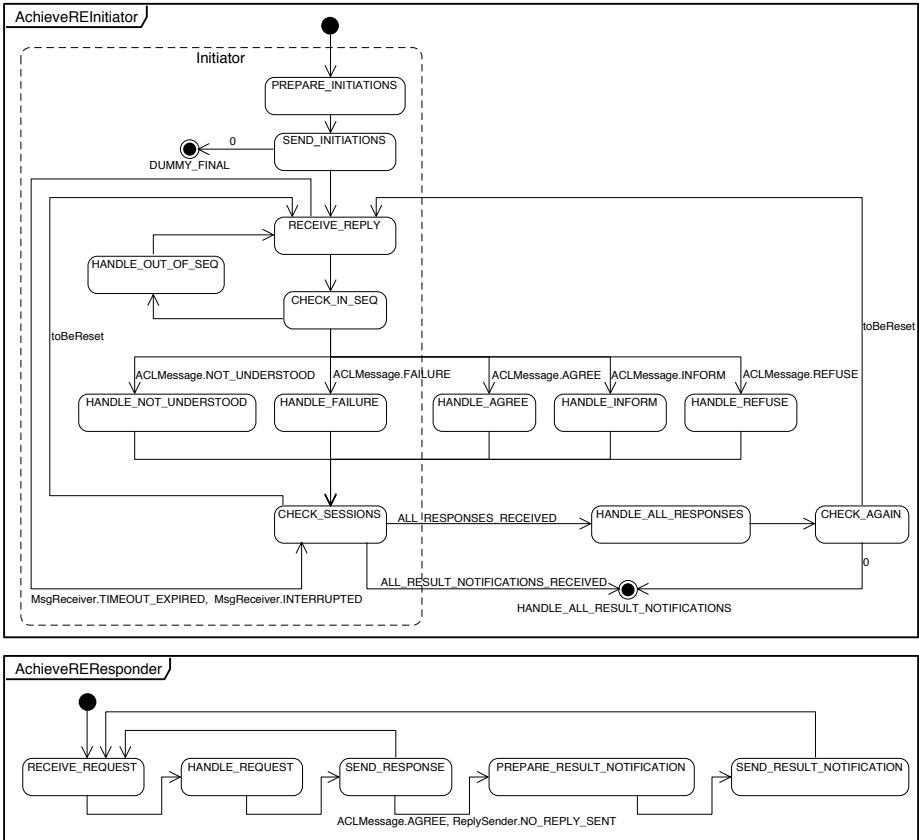


Fig. 1. FIPA-Request IP implemented as FSM

4.2 FIPA-Request IP Implemented as BT

Figure 2 shows our implementation of FIPA-Request IP using *JBehaviourTrees Framework*. *AchieveREInitiatorBT* class implements the same functionality as *AchieveREInitiator* class. It is realized using one *SequentialTask* where an agent forms one or more requests, sends them, and then waits on responses until all are received, or a time-out occurs. Waiting on responses is implemented within one *ParallelTask*, while *DecoratorTask* counts how many messages are received. After one message is received, we use *SelectorTask* in order to determinate type of the response (e.g. failure, not understood, agree). *AchieveREResponderBT* can be used interchangeably with *AchieveREResponder*. It is implemented using one *DecoratorTask* that never ends, and then through one *SequentialTask* is achieved functionality of receiving request and sending response.

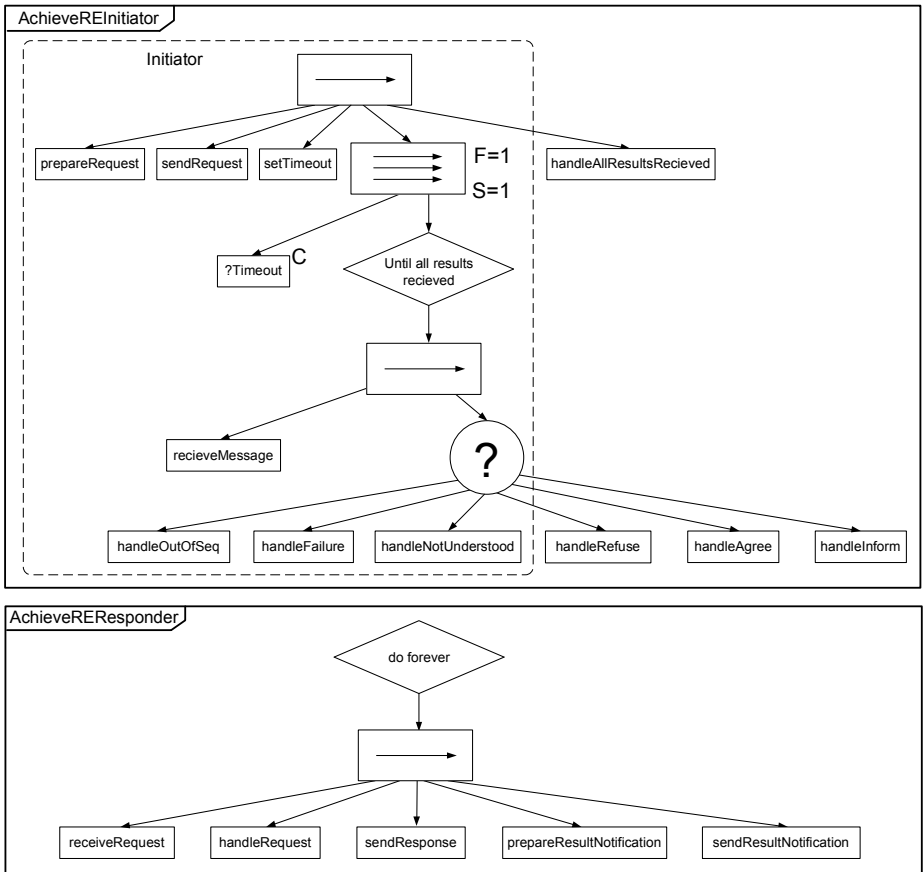


Fig. 2. FIPA-Request IP implemented as BT

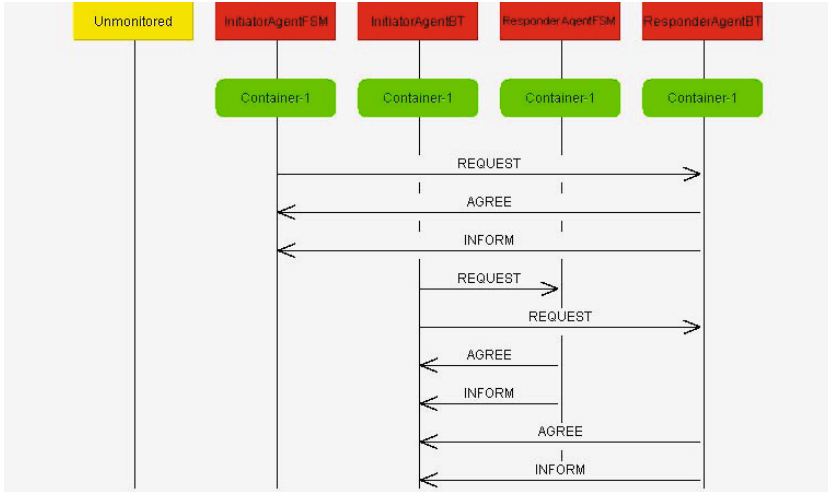


Fig. 3. Message exchange in our version of JADE Sniffer [9]

4.3 Evaluation Results

In order to make functional evaluation of BT implementation in case study, we made three testing sequences: *InitiatorAgentFSM* interacting with *ResponderAgentBT*, *InitiatorAgentFSM* interacting with *ResponderAgentBT* and vice versa. Figure 3 shows messages exchanged in all three scenarios. This verifies the functionality of BT approach and compatibility with build-in FSM implementation.

5 Conclusion

In this paper, we made functional prototype of FIPA-Request Interaction Protocol using proposed *JBehaviourTrees Framework* that provides BTs model for JADE behaviours. We showed that both the FSM implementation and implementation based on BTs can be used interchangeably.

Advantage of our approach is that BT model provides better code reusability. By creating a few simple BTs almost any desired functionality can be achieved by linking trees in different ways or extending the functionality through decorators. This form of creating complex behaviours is more favourable to FSM or even HFSMs due to the large amount of freedom BTs intrinsically support.

However, sometimes it is hard to build BTs that must implement state-machine-like behaviours since that can be only done by creating unintuitive trees. Therefore, for the future work we plan to build a hybrid system, where agents will have multiple BTs and state machines will be used to determine which BT they are currently running.

Acknowledgments. The authors acknowledge the support of research project "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. AiGameDev-site, <http://aigamedev.com/insider/presentations/behavior-trees>
2. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology* 50, 10–21 (2008)
3. FIPA, <http://www.fipa.org>
4. Fortino, G., Rango, F., Russo, W.: Statecharts-Based JADE Agents and Tools for Engineering Multi-Agent Systems. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6276, pp. 240–250. Springer, Heidelberg (2010)
5. Griss, M.L., Fonseca, S., Cowan, D., Kessler, R.: Using UML state machine models for more precise and flexible JADE agent behaviors. In: Giunchiglia, F., Odell, J.J., Weiss, G. (eds.) AOSE 2002. LNCS, vol. 2585, pp. 113–125. Springer, Heidelberg (2003)
6. Heckel, F.W.P., Youngblood, G.M., Ketkar, N.S.: Representational Complexity of Reactive Agents. In: Proceedings of the IEEE Conference on Computational Intelligence and Games, pp. 257–264. IEEE, Los Alamitos (2010)
7. JADE-FSM-Builder, <http://www.sicnet.it/jade-fsm-builder>
8. Jurasovic, K., Jezic, G., Kusek, M.: Using BDI agents for automated software deployment in next generation networks. In: Proceedings of the Eleventh International Conference on Software Engineering and Applications, pp. 423–428. ACTA Press (2007)
9. Kusek, M., Jezic, G.: Extending UML sequence diagrams to model agent mobility. In: Padgham, L., Zambonelli, F. (eds.) AOSE VII / AOSE 2006. LNCS, vol. 4405, pp. 51–63. Springer, Heidelberg (2007)
10. Lim, C.-U., Baumgarten, R., Colton, S.: Evolving Behaviour Trees for the Commercial Game DEFCON. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2010. LNCS, vol. 6024, pp. 100–110. Springer, Heidelberg (2010)
11. Millington, I.: *Artificial Intelligence for Games*. Morgan Kaufmann Publishers, San Francisco (2009)
12. Poggi, A.: Developing multi-user online games with agents. *WSEAS Transactions on Computers* 7, 1240–1249 (2008)
13. Puga, G.F., Gómez-Martín, M.A., Díaz-Agudo, B., González-Calero, P.A.: Dynamic Expansion of Behaviour Trees. In: Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, pp. 36–41. AAAI Press, Menlo Park (2008)
14. Wooldridge, M., Jennings, N.R.: *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review 10, 115–152 (1995)

Scenario Description Language in Multi-agent Simulation System

Armin Stranjak¹, Igor Čavrak², and Mario Žagar²

¹ Software Centre of Excellence, Rolls-Royce plc, PO Box 31, Derby DE24 8BJ, UK

² University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia

armin.stranjak@rolls-royce.com, {igor.cavrak,mario.zagar}@fer.hr

Abstract. Multi-agent system based simulation is used in a number of different areas ranging from modelling of social and economic to technical complex and dynamic systems. In this paper we analyse feasibility of using scenario description language and platform in design and implementation of MAS-based models. A case study is presented where the SDLMAS framework is used to aid in design and implementation of a complex Aero Repair and Overhaul simulator.

Keywords: Multi-Agent System, Interaction, Scenario, Language, Simulation, Aero Repair and Overhaul.

1 Introduction

Multi-agent system (MAS) based simulations are being used in a variety of domains ranging from manufacturing, supply chains, artificial financial markets, electric power markets, modelling social behaviour, sociology and political science. MAS simulations are especially suitable for large scale complex and dynamic systems whose parameters change frequently during run-time which makes analytical methods formidable to use. MAS-based models and simulations efficiently incorporate various resource constraints, dynamic environment, frequently opposing goals of disparate process participants, and ensure satisfying confidence into simulation outcomes. The increased interest into exploitation of agent-based simulations resulted in development of numerous tools and frameworks such as MASON [8], Swarm [9], Repast [10] and NetLogo [13].

MAS-based models are characterised by direct mapping between the entities from the problem (modelled) domain and the model elements representing them. Entities within the target system (simulation) and interactions between them are, in majority of cases, explicitly and individually represented in the model. Such an approach proves beneficial such that: (a) it facilitates the process of transferring concepts and ideas from the problem domain to the modelling (solution) domain, (b) it eases the model validation and verification by (partially) allowing easier mapping of modelled behaviour to particular model components, and (c) it facilitates analysis and interpretation of simulation results to modellers and domain experts.

In this paper we devote a particular interest to a class of MAS-based simulations with properties of (a) having a number of well-defined roles (agent types) characterised by role-specific behaviour and goals, (b) complex behaviours defined by those roles, (c) a moderate to large number of model agents playing one of the defined roles, and (d) complex and explicit interactions.

Such MAS-based simulation systems share a significant portion of design and implementation phase properties with 'regular' multi-agent systems. It is thus a reasonable assumption that such systems can be based on general-purpose multi-agent platforms or frameworks such as JADE [7], Cougaar [3] etc. In order to facilitate such implementations, a number of tools can be used to address various aspects of such approach.

One of the most important, time consuming and error-prone aspects is the design and implementation of complex and explicit interactions among model-constituting roles (and consequently agents playing those roles within a concrete simulation instance). To alleviate the interaction design complexity problem, a scenario-based approach is advocated by a number of researchers such as IOM/T language [4], Q language [6], interaction protocol modelling framework [12], MAP language [17] and others. Mentioned approaches create an explicit representation of interactions between agents with a varying degree of independence from the target agent platform, implementation language and the implementation of concrete role (agent) logic.

The SDLMAS (Scenario Description Language for Multi-Agent Systems) platform [2][14] has been devised in order to support rapid design, development and deployment of multi-agent systems. The declarative language in conjunction with the corresponding development and run-time framework forms the core components of the SDLMAS platform.

In the rest of the paper, Section 2 presents the main properties of the SDLMAS language and platform, characteristics that make SDLMAS suitable for design and implementation of MAS-based simulations. Section 3 presents a case study of SDLMAS application in design and implementation of an Aero Repair and Overhaul simulator. Section 4 concludes the paper.

2 SDLMAS

2.1 Scenario Description Language

The SDLMAS language is a declarative, interaction-centric description language, designed with the purpose of defining permitted sequences of messages (communicative acts) exchanged among interaction participants (agents in a multi-agent system). The language defines an interaction protocol implicitly, as a series of actions where protocol definition is achieved through a sequential approach of agent actions. Conversations among agents are described as a sequence of conversation actions, where actions define a conditional mapping between incoming and outgoing messages, and an agent's internal logic. Conditions for reception and transmission of messages are defined explicitly as a part of a conversation

protocol definition. A procedure is defined as an elementary action of the language, only to be executed as a result of a condition being satisfied following receipt of one or more messages.

A scenario consists of a series of conversation actions describing a conversation protocols amongst agents. The language addresses only the communication aspect of multi-agent systems, and therefore imposes strict division between agent's internal business logic and conversation actions. Developed scenarios are independent of the agent platform and implementation language. This is due to language's declarative nature and strict separation of communication aspects of multi-agent system from implementation of agents' internal business logic.

Agent types or roles, agent references and a set of interaction scenarios are the main elements for conversational behaviour description of multi-agent systems using SDLMAS language. A scenario description has a textual format where the header part contains the definitions of agent types and agent references, while the rest of the body contains a set of scenarios that the system in question needs to perform. An agent role is defined as a standardised pattern of behaviour required by all agents participating in conversations. SDLMAS utilises an implicit approach to role behaviour definitions by defining a number of roles that belong to conversations actions within all defined scenarios.

Conversation actions are defined within the scope of a scenario and are attached to agent roles. Each conversation action defines a procedure and two communication conditions: precondition and postcondition (with exception of first and last scenario action). The procedure represents an internal agent logic function, and is invoked by the SDLMAS runtime framework upon satisfaction of communicative preconditions. Communicative preconditions and postconditions are defined with respect to message performative(s) and message originating agent role(s). A communicative precondition defines circumstances under which a received message or a set of received messages are being passed to an internal procedure implementing agent logic. A condition is defined as a list of expected message performatives and their originating agent roles, and can form expressions using logical operators. Communicative postconditions ensure that all messages generated by agent's internal procedure conform to message transmission conditions.

Agent references correspond to a group of agents attached to a specific role. All references are declared before scenario description definitions of SDLMAS file. They are characterised by their role, cardinality and binding method. There are five agent reference types identified: verbatim, variable, anonymous, role group and variable group references.

2.2 SDLMAS Platform

SDLMAS platform provides tools and a framework for implementation and runtime support of multi-agent systems whose interactions are modelled using SDLMAS language. Several components make a set of core elements of this platform: Management Agent, responsible for system initialisation, configuration and system monitoring; Application agents, developed agents with a support of

SDLMAS language and corresponding code generator; Naming Service Agent, responsible to provide subscribe/publish service for agent discovery; underlying target agent platform components including ORB related libraries, etc.

The SDLMAS platform relies on core functionality provided by the target agent platform such as agent lifecycle management, FIPA compliant messaging, agent container management, naming service as a central storage of agent references, etc. Upon their successful initialization by the Management Agent, all agents are required to register with the Naming Service Agent, and to deregister prior to their deactivation.

Behaviour of an application agent is defined in the SDLMAS scenario description. Programme code generated from the scenario description guarantees conformance to the agent's given role in the system. Code skeletons are generated automatically, while portions of agent code related to the agent's internal logic should be manually developed.

Based on system configuration and scenario descriptions, the Management Agent creates and runs a number of Application agents. After they are initialised and running, Application agents register with the Naming Service and subscribe to other system agents, as determined by the scenario description. Finally, a number of scenarios are initiated by the agents, according to the SDLMAS system scenario descriptions.

2.3 SDLMAS and MAS-Based Simulations

Galan et al. [5] describe a process for designing, implementing and using an agent-based model by identifying four distinct roles and their activities and interactions. This process is used to identify potential points of introducing errors and artefacts. Most of the errors and assumptions are a result of incomplete information passing between process roles and errors in processes of abstraction, approximation or implementation, depending on a role.

SDLMAS framework addresses some of the issues comprising activities performed by a modeller, computer scientist and programmer and interactions between them, thus lowering the risk of miscommunication and implementation errors: (a) it facilitates cross-domain communication (a modeller is seldom from the computing domain) by using a simple declarative description language, thus preventing introduction of interaction-related errors in the process of model approximation; (b) it prevents creation of ambiguous or defect scenario specifications by providing means for formal verification of scenario properties (completeness, liveness etc.); (c) it eliminates errors in the process of model implementation (interaction aspect), drastically reducing implementation time and allowing programmer to focus on business logic aspect of the implementation; and (d) allows easier portability among agent platforms, computer platforms and implementation languages, allowing alternative implementations of the same executable model as a mechanism for detection of artefacts and implementation errors.

However, usage of SDLMAS framework introduces a new type of implicit assumptions related to specifics of runtime semantics; specifically the way multiple

and parallel interactions are handled by specific agent implementations. Such assumptions are handled by metadata specifications determining the execution semantics of scenario actions.

3 Modelling of Aero Engine Scheduling Scenario

With a constant increase of deployed aero engines, Aero Repair and Overhaul (AR&O) industry also needs to scale and improve in order to meet ever-growing need for efficient and effective engine scheduling and servicing. The main challenge lies in meeting the following requirements: (a) timely prediction of engines need for repair in order to satisfy the cost, risk and revenue trade-offs; (b) maximum utilisation of overhaul facilities and minimum of aircraft-on-ground occurrences due to engine unavailability; (c) re-scheduling mechanism to achieve high robustness against unforeseen events that can cause major operation and schedule disruptions; (d) long-term simulated visibility of aero repair and scheduling decisions and major effects of business decisions.

Inherently, AR&O business is greatly versatile that is characterised by large numbers of airline operators, engine fleets, overhaul bases, logistics, etc. that contributes to the non-trivial interaction relationships among them. Such multi-dimensional and complex environment needs to be exploited through modelling and simulations in order to be comprehensively understood and analysed. Simulation of complex systems in the aerospace arena has been proved as a powerful vehicle for rapid prototyping of such systems [1][11], including large-scale research programmes like VIVACE [16]. Simulation of complex systems can be achieved using different approaches such as discrete event modelling, system dynamics, Monte Carlo methods, and finally agent-based simulations. Among these, agents have been shown as the most prominent one with the desirable properties [18].

3.1 Aero Repair and Overhaul System Components

The main functional components of the aero repair and overhaul business are fleet managers, overhaul bases and fleet planner. A Fleet Manager role is, during service, to record engine utilisation and capture engine events such as changes in performance, vibrations, temperature, any sort of irregularities, etc. Statistics such as the number of flights (or cycles) and hours in service, estimation and calculation of reliability of individual critical components and whole-engine reliability is also recorded. A fleet is defined by a set of engines of the same type belonging to a certain airline operator and one particular fleet manager is dedicated to each fleet. An Overhaul Base role is to perform engine repair and maintenance activities. It usually contains several repair lines capable of repairing as many engine types. A Fleet Planner role is to generate overhaul base visit schedules on-demand based on current estimation of remaining engine life. The estimation of a date in the future when an engine needs to be serviced can be determined based on current utilisation rate and allowed number of cycles. It

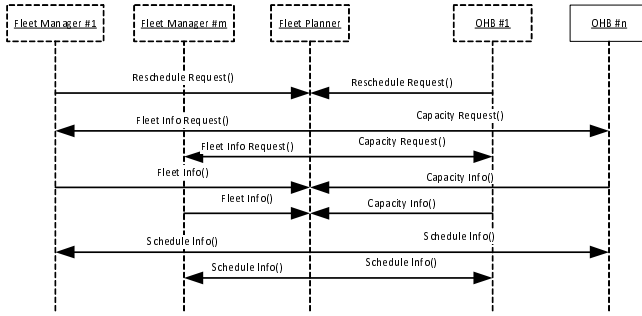


Fig. 1. Rescheduling Scenario

also can be based on a prediction (via continuous monitoring) about when the conditions of engine components reach certain acceptable limits or margins of certain parameters. On top of these, occurrence of an unforeseen event could cause changes of removal date, or indeed induce prompt engine removal and consequent repair. Other factors that the Fleet Planner considers are engine flight pattern and overhaul base capacity, capability and availability. At the same time, the main aim of the Fleet Planner is to maximise revenue by keeping engines in service as long as it is acceptable.

3.2 Scheduling Scenario and Simulator

Scheduling scenario is shown in Figure 1. There are two ways how engine overhaul visit schedule creation can be initiated: (a) on request of fleet managers, or (b) on request of overhaul bases. Either one of these will initiate rescheduling by sending Reschedule Request message.

As soon as Fleet Manager, responsible for certain engine fleet, realises that a particular engine from its fleet needs to be serviced in the period within Schedule Time Horizon (STH) (which is speculative parameter of schedule "visibility" in the future), it will send Reschedule Request message to the Fleet Planner. In other words, if engine's removal date lies within this period, Fleet Manager will request new schedule to be created. The removal date can alter due to: (a) normal engine operation where maximum number of acceptable flights before servicing is estimated to be within STH; (b) changes in an engine's usage pattern or utilisation rate will "bring" engine within STH; (c) fleet-wide updates of engine characteristics will require engines removal in the near future (and within STH); (d) an unforeseen event such as a bird strike that will cause imminent engine removal for servicing.

Overhaul Base would initiate rescheduling operation if there are certain changes in the normal overhaul operation. Typical obstructing events are: (a) delays in current repair operation when engine will not be repaired in time and it potentially can "consume" other engine's repair time; (b) maintenance activities when parts of or entire overhaul base is closed down; (c) unforeseen events such as flood or fire, when normal repair operations are obstructed or cancelled.

```

1 agent @fleetManager : FleetManager;
2 agent @ohb          : OHB;
3 agent fleetPlanner  : FleetPlanner;
4
5 scenario Rescheduling {
6   action FleetManager in rescheduleRequest1() {
7     msgSnd : (RESCHEDULE_REQUEST -> fleetPlanner);
8   }
9
10  action OHB in rescheduleRequest2() {
11    msgSnd : (RESCHEDULE_REQUEST -> fleetPlanner);
12  }
13
14  action FleetPlanner in rcvRescheduleRequest() [timeout=2000]{
15    msgRcv : (RESCHEDULE_REQUEST <- @fleetManager) |
16            (RESCHEDULE_REQUEST <- @ohb);
17    msgSnd : (FLEET_INFO_REQUEST | #BUSY -> <FleetManager>) &
18            (CAPACITY_REQUEST | #BUSY -> <OHB>);
19  }
20
21  action FleetManager in rcvFleetInfoRequest () {
22    msgRcv : (FLEET_INFO_REQUEST | #BUSY <- fleetPlanner);
23    msgSnd : (FLEET_INFO -> fleetPlanner);
24  }
25
26  action OHB in rcvCapacityRequest () {
27    msgRcv : (CAPACITY_REQUEST | #BUSY <- fleetPlanner);
28    msgSnd : (CAPACITY_INFO -> fleetPlanner);
29  }
30
31  action FleetPlanner in generateSchedule () {
32    msgRcv : (FLEET_INFO <- <FleetManager>) &
33            (CAPACITY_INFO <- <OHB>);
34    msgSnd : (SCHEDULE_INFO | #BUSY -> <FleetManager>) &
35            (SCHEDULE_INFO | #BUSY -> <OHB>);
36  }
37 }

```

Fig. 2. Rescheduling SDLMAS scenario definition

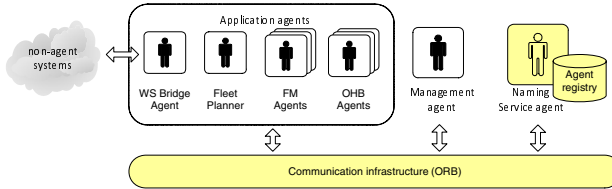


Fig. 3. AR&O Simulator Architecture

After it receives such requests for rescheduling operation, Fleet Planner responds with request for fleet-related and capacity information from fleet managers and overhaul bases (Fleet Info and Capacity requests). Acquired information will enable Fleet Planner to create a new schedule and communicate it to all affected fleet managers and overhaul bases.

The scenario can formally be described in the SDLMAS language, as shown on Figure 2. `FleetPlanner` agent (line 3) is declared using verbatim reference since there is only one fleet planner in the system with already given name (`fleetPlanner`) before runtime. `FleetManager` and `OHB` agent types are used to define roles (line 1-2) of anonymous references to fleet managers and overhaul bases (`@fleetManager` and `@ohb`), whose exact names will be associated with them during runtime. Rescheduling scenario definition (line 5) consists of series of agent actions. A scenario can be initiated by scheduling request from either fleet managers (lines 6-8) or overhaul bases (lines 10-12) to the fleet planner. Upon arrival of scheduling request, fleet planner will request information from

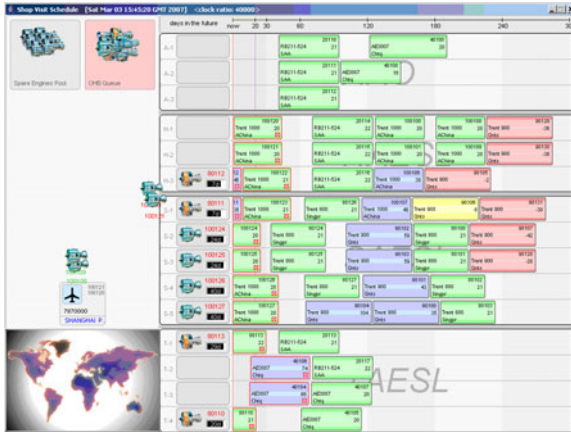


Fig. 4. AR&O Simulator GUI

all fleet managers (`<FleetManager>`) and overhaul bases (`<OHB>`) in order to retrieve fleet-related and overhaul capacity information (lines 14-19). Alternatively, fleet planner can reply with `#BUSY` terminating message performative, which would indicate that cannot deal with the rescheduling request at that time. After receiving request message, fleet managers and overhaul bases will reply with the fleet (line 21-24) and capacity information respectively (lines 26-29). Finally, fleet planner will create a new schedule and communicate it back to the fleet managers and overhaul bases (lines 31-36).

The aero repair and overhaul simulator architecture, based on SDLMAS platform is shown in Figure 3. Apart from `Management` and `Naming Service` agents, which are part of the SDLMAS platform, `FleetPlanner`, `FleetManager` and `OHB` agents are generated automatically with only internal business logic developed manually. The code is generated using code generators from the SDLMAS scenario description into Java code. Code generation details are outside of scope of this paper and more details can be found in [2]. JADE [7] platform was selected as a base platform (which the SDLMAS platform is built on top of) to support agent creation, communication and life-cycle management. The choice of JADE platform was guided by its FIPA compliance, flexible and open architecture and well-established platform among agent community. `WS Bridge Agent` is developed as an interface to live fleet and engine data through Web Services.

The developed simulator (Figure 4) [15] is capable of simulating several decades of simulated time (one simulated day is roughly equivalent to a few seconds of real time) of engine service utilisation and overhaul visits. Such long simulations are necessary since normal engine lifetime can span over several decades. Analysis of multidimensional parameter space and their influence to overall attributes such as revenue stream, costs, availability, quality of service, etc. is highly complex and convoluted task. Engine utilisation, different environment conditions, OHB availability and capability, application of anticipated

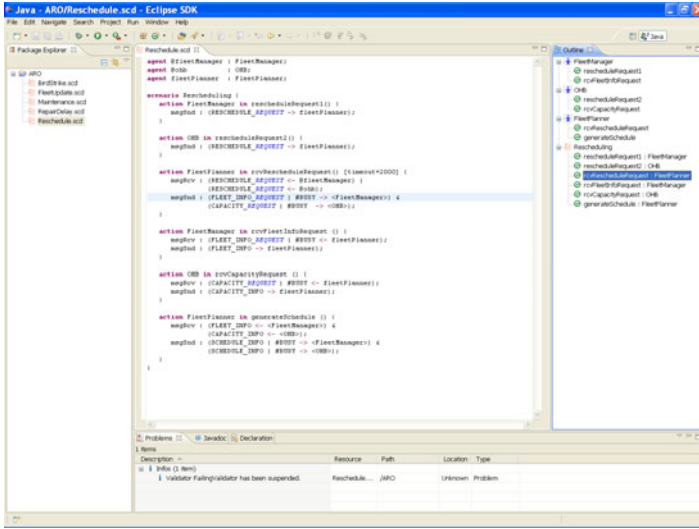


Fig. 5. SDLMAS tool plugin

statistical distribution of unforeseen events, engine reliability and risk estimation models are parameters that have significant influence into the estimation of the future date when an engine needs to be serviced. Only feasible way to analyse this multidimensional space and understand correlations between them is via "what-if" analysis. Such analysis would pre-configure set of parameters and capture interesting system values after every simulation. This way, it is possible, though indirectly, to find correlation matrices between e.g. number of engines in the fleet (including spare engines for quick swaps) and repair turnaround time or OHB utilisation, or number of "aircraft-on-ground" incidents when aircraft cannot fly due to lack of available engines.

The role of SDLMAS language for description of rescheduling scenario and SDLMAS platform are pivotal for the design and development of this simulator. Scenario description helps to design conversational interactions between agents and enables automatic code generation (Figure 5). Majority of code is generated and the developer is required to populate generated code skeletons with agent reasoning logic to make decisions. Furthermore, SDLMAS platform enables entire development and runtime framework for the simulator with minimum configuration and setup tasks. Our estimation is that design and development of the simulator is reduced by factor of 3 if compared with the manual design and development of agent interactions. Also, as a result of usage of code generator, the system is inevitably more robust and stable.

4 Conclusions and Future Work

In this paper we address the field of multi-agent system based simulations, inherent properties found in such systems and focus on a particular subclass of

simulations with implicit and complex interactions among constituent model elements. A SDLMAS framework, consisting of agent scenario description language, tools and a runtime platform is presented and its capabilities in the field of MAS-based simulations analysed. Finally, an agent-based Aero Repair and Overhaul simulator is presented as a proof-of-concept that the SDLMAS framework can successfully address certain issues in design and implementation of complex MAS-based simulations.

The future work on SDLMAS is directed towards enhancing the general capabilities of the language and platform, primarily in enhancement of control over runtime conversation context handling (scenario loops, context concatenation and nesting) and runtime semantics (re-entrant scenario actions, parallel conversation handling).

References

1. Agarwal, R., Gilmer, J.: Back from the future: Using projected futures from recursive simulation to aid decision making in the present. In: Proceedings of the International Symposium on Military, Government and Aerospace Simulation (2004)
2. Cavrak, I., Stranjak, A., Zagar, M.: SDLMAS: A Scenario Modelling Framework for Multi-Agent Systems. *Journal of the UCS* 15(4), 898–925 (2009)
3. Cougaar: Cognitive Agent Architecture, <http://www.cougaar.org/>
4. Doi, T., Tahara, Y., Honiden, S.: IOM/T: An Interaction Description Language for Multi-Agent Systems. In: 4th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 778–785. ACM, New York (2005)
5. Galán, J.M., Izquierdo, L.R., Izquierdo, S.S., Santos, J.I., del Olmo, R., Lopez-Paredes, A., Edmonds, B.: Errors and Artefacts in Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation* 12(1) (2009)
6. Ishida, T.Q.: A Scenario Description Language for Interactive Agents. *IEEE Computer* 35(11) (2002)
7. Java Agent Development Framework, <http://jade.cselt.it/>
8. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A multi-agent simulation environment. *Simulation* 81, 517–527 (2005)
9. Minar, C.L.N., Burkhart, R., Askenazi, M.: The swarm simulation system: A toolkit for building multi-agent simulations. Tech. report, Sante Fe Institute (1996)
10. North, M.J., Collier, N.T., Vos, J.R.: Experiences creating three implementations of the Repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation* 16(1), 1–25 (2006)
11. Pritchett, A.R., van Paassen, M.M., Wieland, F.P., Johnson, E.N.: Aerospace vehicle and air traffic simulation. In: *Applied System Simulation: Methodologies and Applications*, pp. 365–389. Kluwer Academic, Dordrecht (2003)
12. Quenum, J.G., Aknine, S., Briot, J.-P., Honiden, S.: A modeling framework for generic agent interaction protocols. In: Baldoni, M., Endriss, U. (eds.) *DALT 2006*. LNCS (LNAI), vol. 4327, pp. 207–224. Springer, Heidelberg (2006)
13. Sklar, E.: Software Review: NetLogo, a Multi-agent Simulation Environment. *Artificial Life* 13, 303–311 (2007)
14. Stranjak, A., Čavrak, I., Žagar, M.: Scenario Description Language for Multi-agent Systems. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) *IEA/AIE 2008*. LNCS (LNAI), vol. 5027, pp. 855–864. Springer, Heidelberg (2008)

15. Stranjak, A., Dutta, P.S., Ebden, M., Rogers, A., Vytelingum, P.: A Multi-Agent Simulation System for Prediction and Scheduling of Aero Engine Overhaul. In: Proc. 7th International Conference on Autonomous Agents and Multiagent Systems: industrial track (AAMAS 2008), Estoril, Portugal, pp. 81–88 (2008)
16. Value Improvement through a Virtual Aeronautical Collaborative Enterprise (VIVACE), <http://www.vivaceproject.com>
17. Walton, C.D.: Multi-Agent Dialogue Protocols (2003)
18. Yu., T.-T.: A Comparison of Agent-based Modelling with Traditional Discrete-event Modelling. Report ECSTR-LSL07-006, University of Southampton (2007)

Analyzing the Validity of Passive Investment Strategies Employing Fundamental Indices through Agent-Based Simulation

Hiroshi Takahashi¹, Satoru Takahashi², and Takao Terano³

¹ Graduate School of Business Administration, Keio University,
4-1-1 Hiyoshi, Kohoku-ku, Yokohama-city, 223-8572, Japan

² Chuo Mitsui Asset Trust and Banking Company, Limited, Japan

³ Tokyo Institute of Technology, 4259 Nagatsuda, Midori-ku, Yokohama, Japan
htaka@kbs.keio.ac.jp, satoru_1.takahashi@chuomitsui.jp,
terano@dis.titech.ac.jp

Abstract. This research analyzes the influence of indices which are employed in the asset management business on financial markets through agent-based simulation. In this analysis, we focus on a fundamental index, which has been proposed as a new benchmark for investments in place of price indices, which are currently employed in practical business affairs. As a result of intensive experiments in the market, we made the following interesting findings: (1) fundamental indexing works as effectively as a price indexing in the market when market prices reflect fundamental values; (2) fundamental indexing contributes to market efficiency. However, we also found drawbacks to fundamental indexing, such as the risk of destabilizing markets when too many investors employ passive investment strategies using the fundamental index. These results are significant from both practical and academic viewpoints. These analyses also demonstrate the effectiveness of agent-based techniques for financial research.

Keywords: Agent Based-Model, Financial Engineering, Behavioral Finance, Passive Investment Strategy, Fundamental Indexation.

1 Introduction

Understanding the mechanisms of financial markets is critical because of the rapid globalization of financial markets and the rigidity of economic systems. Market Efficiency is one of the central concerns of financial research and a great deal of research regarding market efficiency has been conducted. In efficient markets, it is difficult to achieve excess return compared to the return which might reasonably be expected from the risk involved. With this in mind, passive investment strategies that follow market indices such as TOPIX and S&P500 are conventionally regarded as effective investment methods. On the other hand, there are many analyses - such as those of behavioral financial theory - which throw doubt on the traditional asset-pricing hypothesis that markets operate

efficiently. While it is true that passive investment strategies are valid and effective in efficient market, it is unclear if this is also the case in markets that contain inefficiencies [6] [8] [7].

Considering these factors, a fundamental index was proposed as a substitute for price indexing in the asset management business [1]. A fundamental index is calculated based on the value of a company's financial statements - profit and so on - instead of the market price that is commonly used in business affairs. Compared to price indexing, fundamental indexing is relatively unaffected by the decision bias of investors and could have better characteristics than price indexing. Additionally fundamental indexing could actually contribute to market efficiency. Therefore, the analysis of fundamental indexing is significant from both academic and practical points of view.

The purpose of this research is to clarify the influence of fundamental indexing on the market through agent-based simulation. In the case of financial markets, investors represent agents and a stock market represents a multi agent system. In this study, we investigate the following issues: (1) the validity of fundamental indexing; and (2) the influence of fundamental indexing on financial markets. The next section describes the model used in this analysis. Section 3 shows the results of the analysis. Section 4 summarizes this paper.

2 Design of the Artificial Financial Market

A computer simulation of the financial market involving 1000 investors was used as the model for this research. Shares and risk-free assets were the two types of assets used, along with the possible transaction methods. Several types of investors exist in the market, each undertaking transactions based on their own stock evaluations. This market was composed of three major stages: (1) generation of corporate earnings; (2) formation of investor forecasts; and (3) setting transaction prices. The market advances through repetition of these stages. The following sections describe negotiable transaction assets, modeling of investor behavior, setting of transaction prices, and the rule of natural selection.

2.1 Negotiable Assets in the Market

This market has risk-free and risk assets. There are risky assets in which all profits gained during each term are distributed to the shareholders. Corporate earnings (y_t) are expressed as $y_t = y_{t-1} \cdot (1 + \varepsilon_t)$. However, they are generated according to the process of $\varepsilon_t \sim N(0, \sigma_y^2)$ with share trading being undertaken after the public announcement of profits for the term. Each investor is given common asset holdings at the start of the term with no limit placed on debit and credit transactions (1000 in risk-free assets and 1000 in stocks). Investors adopt the buy-and-hold method for the relevant portfolio as a benchmark to conduct decision-making by using a one-term model.

2.2 Modeling Investor Behavior

Each type of investor considered in this analysis is organized in Table 1. Type 1-4 are active investors and type 5-6 are passive investors. This section describes the investors' behavior.

Passive Investor. This section describes the behavior of passive investors who try to follow a price index or a fundamental index. Passive investors automatically decide their investment ratio in stock based on the index's value. Passive investors who try to follow the price index employ a buy-and-hold strategy in this market. Passive investors who try to follow a fundamental index decide their investment ratio in stock based on the value of the fundamental index.

Active Investors. Active investors in this market evaluate transaction prices based on their own market forecasts, taking into consideration both risk and return rates when making decisions. Each investor determines the investment ratio (w_t^i) based on the maximum objective function($f(w^i_t)$), as shown below.

$$f(w_t^i) = r_{t+1}^{int,i} \cdot w_t^i + r_f \cdot (1 - w_t^i) - \lambda(\sigma_{t-1}^{s,i})^2 \cdot (w_t^i)^2. \tag{1}$$

Here, $r_{t+1}^{int,i}$ and $\sigma_{t-1}^{s,i}$ expresses the expected rate of return and risk for stocks as estimated by each investor i . r_f indicates the risk-free rate. w_t^i represents the stock investment ratio of the investor i for term t . The investor decision-making model here is based on the Black/Litterman model that is used in securities investment [4]. The expected rate of return for shares is calculated as follows.

$$r_{t+1}^{int,i} = (r_{t+1}^{f,i}c^{-1}(\sigma_{t-1}^i)^{-2} + r_t^{im}(\sigma_{t-1}^i)^{-2}) / (1 \cdot c^{-1}(\sigma_{t-1}^i)^{-2} + 1 \cdot (\sigma_{t-1}^i)^{-2}). \tag{2}$$

Here, $r_{t+1}^{f,i}$ $C r_t^{im}$ expresses the expected rate of return, calculated from the short-term expected rate of return, plus risk and gross price ratio of stocks respectively. c is a coefficient that adjusts the dispersion level of the expected rate of return calculated from risk and gross current price ratio of stocks [4].

The short-term expected rate of return ($r_t^{f,i}$) is obtained where ($P_{t+1}^{f,i}, y_{t+1}^{f,i}$) is the equity price and profit forecast for term $t + 1$ is estimated by the investor, as shown below:

Table 1. List of investors type

No.	Investor type
1	fundamentalist
2	forecasting by latest price
3	forecasting by trend(most recent 10 days)
4	forecasting by past average(most recent 10 days)
5	passive investor(price index)
6	passive investor(fundamental index)

$$r_{t+1}^{f,i} = ((P_{t+1}^{f,i} + y_{t+1}^{f,i})/P_t - 1) \cdot (1 + \eta_t^i). \tag{3}$$

The short-term expected rate of return includes the error term ($\eta_t^i \sim N(0, \sigma_n^2)$) reflecting that even investors using the same forecast model vary slightly in their detailed outlook. The stock price ($P_{t+1}^{f,i}$), profit forecast ($y_{t+1}^{f,i}$), and risk estimation methods are described in section 2.2.

The expected rate of return obtained from stock risk and so forth is calculated from stock risk (σ_{t-1}^i), benchmark equity stake (W_{t-1}), investors' degree of risk avoidance (λ), and risk-free rate (r_f), as shown below 4.

$$r_t^{im} = 2\lambda(\sigma_{t-1}^s)^2 W_{t-1} + r_f. \tag{4}$$

Stock Price Forecasting Method. The fundamental value is estimated by using the dividend discount model, which is a well known model in the field of finance. Fundamentalists estimate the forecasted stock price and forecasted profit from the profit for the term (y_t) and the discount rate (δ) as $P_{t+1}^{f,i} = y_t/\delta, y_{t+1}^{f,i} = y_t$.

Forecasting based on trends involves forecasting the following term's stock prices and profit through extrapolation of the most recent stock value fluctuation trends. The following term's stock price and profit is estimated from the most recent trends of stock price fluctuation (a_{t-1}) from time point $t - 1$ as $P_{t+1}^{f,i} = P_{t-1} \cdot (1 + a_{t-1})^2, y_{t+1}^{f,i} = y_t \cdot (1 + a_{t-1})$.

Forecasting based on past averages involves estimating the following term's stock prices and profit based on the most recent average stock value.

Risk Estimation Method. In this analysis, each investor estimates risk from past price fluctuations. Specifically, stock risk is estimated as $\sigma_{t-1}^i = \sigma_{t-1}^h$ (common to each investor). Here, σ_{t-1}^h represents the stock volatility that is calculated from price fluctuation from the most recent 100 terms.

2.3 Determination of Transaction Prices

Transaction prices are determined as the price where stock supply and demand converge ($\sum_{i=1}^M (F_t^i w_t^i)/P_t = N$). In this case, the total asset (F_t^i) of investor i is calculated from transaction price (P_t) for term t , profit (y_t) and total assets from the term $t - 1$, stock investment ratio (w_{t-1}^i), and risk-free rate (r_f), as $F_t^i = F_{t-1}^i \cdot (w_{t-1}^i \cdot (P_t + y_t)/P_{t-1} + (1 - w_{t-1}^i) \cdot (1 + r_f))$.

2.4 Rules of Natural Selection in the Market

The rules of natural selection can be identified in this market. The driving force behind these rules is cumulative excess profit 5. The rules of natural selection go through the following two stages: (1) the identification of investors who alter their investment strategy, and (2) the actual alteration of investment strategy 9.

Each investor determines the existence of investment strategy alteration based on the most recent performance of each 5 term period after 25 terms have passed since the beginning of market transactions. The higher the profit rate obtained most recently is, the lesser the possibility of strategy alteration becomes. The

lower the profit, the higher the possibility becomes. Specifically, when an investor cannot obtain a positive excess profit for the benchmark portfolio, they are likely to alter their investment strategy with the probability below :

$$p_i = \min(1, \max(0.5 \cdot e^{-r_i^{cum}} - 0.5, 0)). \tag{5}$$

Here, however, r_i^{cum} is the cumulative excess profitability for the most recent benchmark of investor i . Measurement was conducted for 5 terms, and the cumulative excess profit was converted to a period of one term.

When it comes to determining new investment strategies, an investment strategy that has had a high cumulative excess profit for the most five recent terms (forecasting type) is more likely to be selected. Where the strategy of the investor i is z_i and the cumulative excess profit for the most recent five terms is r_i^{cum} , the probability p_i that z_i is selected as a new investment strategy is given as $p_i = e^{(a \cdot r_i^{cum})} / \sum_{j=1}^M e^{(a \cdot r_j^{cum})}$. Those investors who altered their strategies make investments based on the new strategies after the next step.

3 Experimental Result

First of all, the validity of a passive investment strategy is analyzed. Afterwards, the conditions for efficient markets are explored through inverse simulation analysis. The examples given here represent the overall nature of our results and are not atypical as a sample.

3.1 Validity of Passive Investment Strategy

This section analyzes the validity of a passive investment strategy using price indexing and fundamental indexing. Firstly, this section analyzes the influence of passive investor employing price indexing on the market; secondly, we analyze the influence of passive investor employing fundamental indexing.

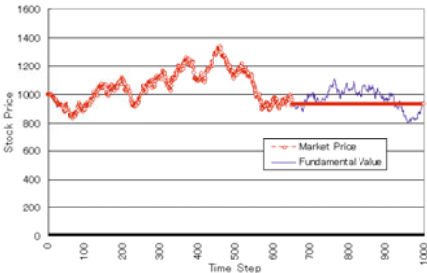


Fig. 1. Price transitions(Price index)

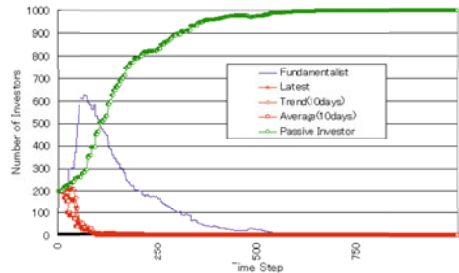


Fig. 2. Transition of the number of investors(Price index)

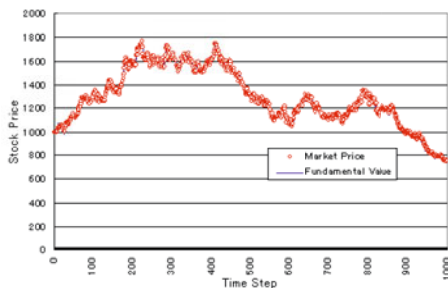


Fig. 3. Price transitions (Fundamental index)

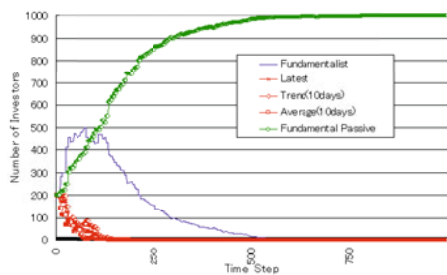


Fig. 4. Transition of the number of investors (Fundamental index)

Validity of Passive Investment Strategy using Price Index. This section analyzes the influence of passive investors using price indexing. Fig. 1 and Fig. 2 show the typical transition of market price and the number of investors with investors categorized into 5 types (Table 1). These results suggest that the market situation can be divided into 3 periods, as follows: (1) the period when the number of fundamentalists increase; (2) the period when the number of investors decreases; (3) the period when all investors employ passive investment strategies using a price index.

During the period (1), the number of fundamentalists and passive investors is increasing. At the same time, the numbers of other types of investors are decreasing. When market prices are consistent with fundamental values, the behavior of passive investors and fundamentalists is the same when averaged out. Therefore, the average performance of both passive investors and fundamentalists is basically the same. Under conditions where five types of investors exist within the market, it is observed that capital tends to flow from non-fundamentalist/non-passive investors to fundamentalist/passive investors. These results support the validity of a passive investment strategy and coincide with traditional asset-pricing theory. During the period (2), only two types of investors, fundamentalist and passive investor, exist in the market. Although the behavior of passive investors and fundamentalists is the same (when averaged out), the predictions of fundamentalists contain error terms. Therefore investors' performance can sometimes fall short of the benchmark, which can prompt them to change their investment strategy in accordance with the natural selection. On the other hand, passive investors' performance is always in line with the benchmark, so they never change their strategy. During the period (3), market prices don't reflect fundamental values, because passive investors doesn't affect market prices at all.

These results suggest that a passive investment strategy is valid. However, they also show that such a strategy can have an unfavorable effect on the market.

¹ These results coincide with traditional financial theory and suggest the validity of this model.

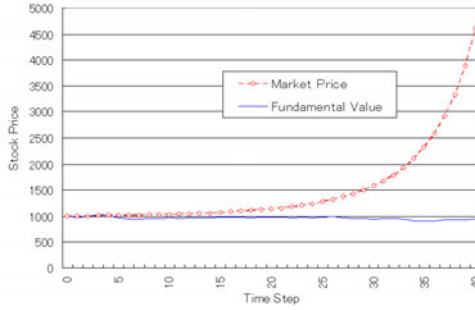


Fig. 5. Price transitions(Trend chaser 1%)

The validity of Passive Investment Strategies using a Fundamental Index. Fig. 3 and Fig. 4 show the transition of market prices and the number of investors where there are the same number of 5 types of investors (Table 1 FType1-4,6), including passive investors using a fundamental index. Market situations can also be categorized into 3 periods, as before. However, market prices are consistent with fundamental values even where all investors employ a passive investment strategy, in contrast to the previous case. These results suggest the following: (1) a passive investment strategy using a fundamental index is as valid as one based on a price index, and (2) compared to the previous results, the influence of passive investors on asset prices increased.

Fig. 5 shows stock price transitions where 1 % of investors at term 800 in Fig 4 turned to trend chasing. It can be seen that market prices deviated significantly from fundamental values. This result suggests that a spike in the number of passive investors could lead to market instability.

These results suggest that although fundamental indexing reduces the drawbacks associated with passive investment strategies which employ a price index - which is common in the asset management business - there still remain some drawbacks to a passive investment strategy.

Analyses with taking a variety of Investor Behavior and Costs into account. This section analyzes and attempts to account for practical market conditions. In particular, this research takes into account a variety of investor behavior and costs. This section considers the following aspects: (1) with respect to costs, the transaction cost is 20 bp/200 terms; (2) the cost for investigation is 20 bp/200 terms²; (3) as for the range of investor behavior, each investor has an average percentage of 1 % possibility to randomly change their strategy³ [9].

The right side of table. 2 shows the average number of investors at the final term. Almost the same number of both types of passive investors exists in the

² Investors other than passive investors using a price index must bear the costs of accessing and setting up a fundamental index.

³ This mechanism corresponds to a mutation in a genetic algorithm [5].

market, and a certain number of fundamentalists also exist in the market⁴. Fundamentalists get an excess return from the market because there is a variety of investors in the market. Therefore, fundamentalists can survive, and a certain number of investors exists in the market⁵. The mechanism for fundamentalists to survive in the market may contribute to market efficiency. It is therefore reasonable to assume that the presence of a variety of investors plays an important role in creating the necessary conditions for the survival of fundamentalists.

Table 2. List of Investor Numbers

The number of investors (at 1000 term)	When both passive investors coexist		When a variety of investors and cost are considered	
	mean	std.	mean	std.
Fundamentalist	0.03	0.22	36.24	46.75
Latest Price	0.00	0.00	5.75	4.02
Historical trend	0.33	2.32	10.95	22.35
Historical average	0.04	0.24	13.48	20.83
Passive (price index)	480.80	330.69	472.13	377.21
Passive(fundamental index)	518.80	330.73	461.45	389.90

Note: This table shows the average and standard deviations of the number of investors during the final term, when simulation with 1000 terms were repeated 100 times.

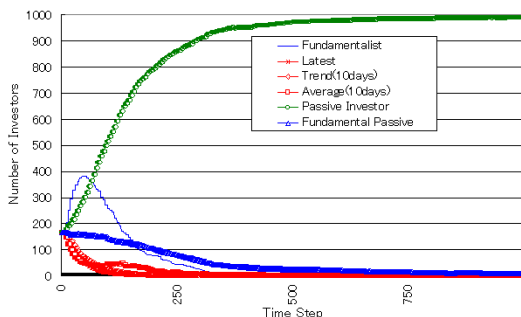


Fig. 6. Transition of the number of investors (investigation cost $\times 10$, selection bias $\times 2$)

When the numbers of both types of passive investors are compared, the number of passive investors using a price index is slightly higher when it comes to using a fundamental index in real markets, investors have several alternatives, such as: (1) calculating a fundamental index by themselves, or; (2) paying the cost for a fundamental index produced by another financial institution.

⁴ The analyses were repeated 100 times and the average number of investors of each type was calculated.

⁵ A variety of investors might play a role in providing capital to fundamentalists in this case.

Whatever the case, fundamentalists have to bear the cost for their fundamental index directly or indirectly. On the other hand, a price index is calculated from price information which is relatively easy to access - and is therefore less costly than a fundamental index. On this point, price indexing has an advantage over fundamental indexing.

Fig. 6 shows the transition of the average number of investors under extreme conditions, where the cost for investigation is ten times higher (200bp/200terms), and the selection bias doubles⁶. Because we focus on the influence of cost on markets in this section, we analyze cases where investors don't change their strategy in a random manner⁷. From the figure, it is confirmed that, as time steps go by, the number of passive investors using a price index gradually increases, while the number of passive investors using a fundamental index decreases⁸. Regarding the transition of market prices, it is observed that deviations from fundamental value emerge in the market(Fig. 11). Although these analyses were carried out under rather extreme conditions, these results suggest that costs associated with accessing/setting up a fundamental index could have a significant influence on financial markets through investors' behavior.

This section analyzes the case where some practical aspects are taken into account. This aspect of our research proved fruitful and further work regarding costs and index accuracy is planned for the future.

4 Summary

This research analyzes the influence of fundamental indexing on markets and made the following findings, (1) fundamental indexing is as effective as price indexing in markets where market prices reflect fundamental values; (2) fundamental indexing could contribute to improved efficiency in markets⁹. This research has attempted to tackle the problem that practical asset management business faces, and has produced results which may prove significant from both academic and practical points of view. These analyses also demonstrate the effectiveness of agent-based techniques for financial research. More detailed analyses relating to cost, market design and several other matters of interest are planned for the future.

⁶ Calculation of the average number of investors are based on 100 simulations. The numbers of investors of Latest, Trend(10days), Average(10days) are almost the same, so it's difficult to distinguish between the different three curves in Fig. 6.

⁷ Although investors don't change their strategy in a random manner, the rules of natural selection still apply in this analysis as with previous analyses.

⁸ These results reconfirm the validity of a passive investment strategy using a price index. On the other hand, analyses suggesting the advantage of a fundamental index over price index based on empirical data have also been reported[11]. Further research is planned.

⁹ Inverse simulation analysis supports conclusion (2), although the results are not included here.

References

1. Arnott, R., Hsu, J., Moore, P.: Fundamental Indexation. *Financial Analysts Journal* 61(2) (2005)
2. Arthur, W.B., Holland, J.H., Lebaron, B., Palmer, R.G., Taylor, P.: Asset Pricing under Endogenous Expectation in an Artificial Stock Market. In: *The Economy as an Evolving Complex System II*, pp. 15–44. Addison-Wesley, Reading (1997)
3. Axelrod, R.: *The Complexity of Cooperation -Agent-Based Model of Competition and Collaboration*. Princeton University Press, Princeton (1997)
4. Black, F., Litterman, R.: Global Portfolio Optimization. *Financial Analysts Journal*, 28–43 (September–October 1992)
5. Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
6. Sharpe, W.F.: Capital Asset Prices: A Theory of Market Equilibrium under condition of Risk. *The Journal of Finance* 19, 425–442 (1964)
7. Shiller, R.J.: *Irrational Exuberance*. Princeton University Press, Princeton (2000)
8. Shleifer, A.: *Inefficient Markets*. Oxford University Press, Oxford (2000)
9. Takahashi, H., Terano, T.: Agent-Based Approach to Investors' Behavior and Asset Price Fluctuation in Financial Markets. *Journal of Artificial Societies and Social Simulation* 6(3) (2003)
10. Takahashi, H., Takahashi, S., Terano, T.: Analyzing the Influences of Passive Investment Strategies on Financial Markets via Agent-Based Modeling. In: Edmonds, B., et al. (eds.) *Social Simulation Technologies: Advances and New Discoveries*, Idea Group Inc., USA (2007)

A List of Main Parameters

This section lists the major parameters of the financial market designed for this paper. Explanations and values for each parameter are described.

M: Number of investors (1000)

N: Number of shares (1000)

F_t^i : Total asset value of investor i for term t ($F_0^i = 2000$: common)

W_t : Ratio of stock in benchmark for term t ($W_0 = 0.5$)

w_t^i : Stock investment rate of investor i for term t ($w_0^i = 0.5$: common)

y_t : Profits generated during term t ($y_0 = 0.5$)

σ_y : Standard deviation of profit fluctuation ($0.2/\sqrt{200}$)

δ : Discount rate for stock (0.1/200)

λ : Degree of investor risk aversion (1.25)

σ_n : Standard deviation of dispersion from short-term expected rate of return on shares (0.01)

a : Degree of selection pressure (20)

c : Adjustment coefficient (0.01)

Dynamic World Model with the Lazy Potential Function

Konrad Kułakowski and Tomasz Stepień

Institute of Automatics,
AGH University of Science and Technology
Al. Mickiewicza 30,
30-059 Cracow, Poland
konrad.kulakowski@agh.edu.pl

Abstract. One of the fundamental skills of an autonomous mobile robot is its ability to determine a collision-free path in a dynamically changing environment. To meet this challenge, robots often have their own world model - an internal representation of the environment. Such a representation allows them to predict future changes to the environment, and thus to plan further moves and actions.

This paper presents an agent-oriented dynamic world model built on top of asynchronous non-homogeneous cellular automaton, equipped with the new collision-free path finding algorithm based on a lazy potential field function. The presented abstract model is preliminarily verified using a specially designed middleware library supporting cellular model simulations.

1 Introduction

The notion of a *world model (WM)* is inextricably connected with a hierarchical planning and control paradigm of autonomous mobile robot architecture [4]. One of the most well-known examples of such an approach is “*A Reference Model Architecture For Unmanned Vehicle Systems*”, which was originally proposed by Albus [1,2]. This assumes that the *WM* is “*an internal representation of the world*”, which may include “*models of objects, events, classes, tasks, and agents; models of portions of the environment*” as well as different kinds of knowledge distributed throughout all the parts of the system. The main purpose of the gathered knowledge is to represent all the important facts from the surrounding world that may facilitate planning, prediction and control. The *WM* may also contain elements of spatial information “*geometric entities*” corresponding to actual objects in the environment. The presented descriptive definition of *WM* is interesting since it seems to be both one of the most accurate and capacious in the literature.

The *WM* is useful, although it is not mandatory. There are a lot of successful constructions which do not use the *WM*. According to Brook’s postulate “*The world is its own best model*” [8], robots may just sense and react to the changes in the environment, and do not use the *WM* in the Albus sense at all. An

example of such an approach is *Subsumption Architecture* [7], in which system control is distributed among behavior schemes on different levels of abstraction. Since a purely reactive approach, besides its benefits, also has several potential shortcomings [4], almost all the contemporary constructions are, in fact, hybrid i.e. they contain *WM*-based deliberative processing as well as real-time behavior driven routines [2,10,3].

There are a lot of different *WM* constructions. Since the *WM* is usually built on the top of a knowledge base (KB) [2], there may be as many different *WM* implementations as knowledge representations. Many of them are based on first order logic as a very popular form of the knowledge representation scheme in *AI* [16]. Thus, the *KB* is often implemented on top of a system supporting logical reasoning. The following four main classes of such a system can be found in the literature [16]:

- theorem provers and logic programming languages
- production systems
- frame systems and semantics networks
- description logic systems

Theorem provers are used to prove formally written sentences (theorems). They use inference procedure (e.g. the resolution rule) to answer the question of whether the proposed formula is satisfiable. The logic programming languages (e.g. Prolog), besides their inference ability, also provide the programmer with non-logical features, such as graphical user interface or input/output library. In the production systems, the primary representation of knowledge are implications. The consequences of implications are interpreted as action proposals. Production systems are based on a forward-chaining paradigm. Frame systems and semantics networks (*SN*) organize knowledge in the form of a graph, where nodes represent objects, and edges between the nodes correspond to binary relations between objects. An example of the use of *SN* in a mobile robotics can be found in [13]. The description logic systems evolved from semantics networks. They model concepts, roles, individuals and relationships between them in a formal way, thus they provide logical formalism for *SN*.

In the paper, the agent-oriented *Dynamic World Model (DWM)* is presented. It is defined on top of asynchronous, non-homogeneous cellular automaton, which is treated as a kind of concurrent production system. The rules encoded into the automaton provide indications of the future behavior of a robot's environment and may suggest future actions of the robot. The presented formal model of the world is accompanied by a collision-free path finding algorithm in the dynamically changing environment. The early implementation of the simulation library, as well as a description of preliminary tests is also included.

2 Motivation

Using as much knowledge about the environment as possible is one of the important challenges of intelligent robotics [17]. Humans have tremendous knowledge

about the objects in their surroundings, but it is not clear how that knowledge can be used by robots. There are numerous works that propose collision-free path finding algorithms [14]. Most of them assume that a model is either static or its dynamics are limited to changes captured from the environment. Thus, despite the object is being recognized and identified by the robot, the robot can not use this knowledge to foresee further states of the model. For instance, a person seeing an adult man crossing his way from the right to the left would probably move to the right side to avoid the man. That is because he predicts that, in a moment, the man will move more to the left. He does not need to observe the man all the time. It is enough to identify the direction of the walk, and the fact that this is an adult. Knowing the typical behavior of an adult the person trusts (knows) that he will continue to follow his way. The idea behind the use of a cellular-based *DWM* is to propose a knowledge model which allows short term prediction of a robot's environment behavior. In particular, representation of the *DWM* as a cellular automaton facilitates the finding of a collision-free path leading to the specified target.

3 Dynamic World Model

The concept of a *Dynamic World Model* is based on *ECAL* (*Extended Non Homogeneous Cellular Automaton*) [9], given as a seven-tuple:

$$ECAL \stackrel{df}{=} (C, R_p, \eta, S, CON, r_{ext}, f) \quad (1)$$

where: C – set of cell lattice, R_p – proximity relation, η – neighborhood function, S – set of cell states, CON – set of cell configurations, r_{ext} – local rule, f – actualization order sensitive state change function. The *ECAL* definition allows us to define a *Dynamic World Model* as a following quintuple:

$$DWM \stackrel{df}{=} (A^*, b_a, O, f_p, ECAL) \quad (2)$$

where A^* is a multiset of agents, $b_a : A \rightarrow S$ is a function, which binds the state of the cell with the type of agent which is assigned to the cell, $O \subset C$ is a multiset of permanent objects (e.g. walls), $f_p : C \rightarrow \mathbb{N} \cup \{0, \infty\}$ is a potential function. A detailed, formal explanation of the *DWM* can be found in [12]. f_p assigns to every cell the value which describes its Manhattan distance [16] to the nearest attractor (potential source). The value of the attractor's potential is 0. Thus, the agent who wants to reach the goal (attractor) in a collision free manner has to follow decreasing values of potential (Fig. 1a). For further convenience, let us define a localization function $l_a : A \rightarrow C$ defining the current location of the given agent in the *DWM*.

It is convenient to distinguish one agent $a_{self} \in A^*$ which identifies the robot. Agent a_{self} is considered as a point and occupies exactly one cell. It is assumed that a_{self} has a map (i.e. it knows the topology of the environment), and knows its position on the map. Thus, it is able to compute the f_p and use it in order to take a decision what to do next. Behavior of the other agents are modeled using

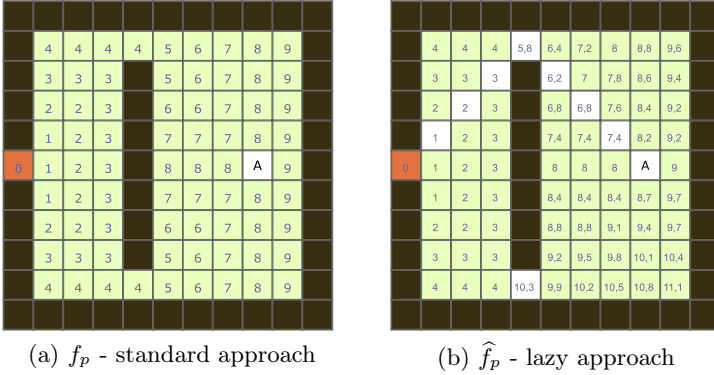


Fig. 1. Values of artificial potential field. In the lazy approach, the path from the agent to the exit leads through the gate, which often remains open

the rules encoded into *ECAL* automaton. Although time dimension is present in the *DWM*, it is hidden in *ECAL*. To make them more explicit, let us define the *TDWM* i.e. *Temporal Dynamic World Model*:

$$TDWM \stackrel{df}{=} (A^*, b_a, O, f_p, T, ECAL) \tag{3}$$

where $T \subset \mathbb{R}$ is an additional discrete set of time points. Let the state of the *TDWM* for the given time point t_x be defined as:

$$state(TDWM, t_x) \stackrel{df}{=} (f_p^{(t_y)}, con^{(t_z)}) \tag{4}$$

where $f_p^{(t_y)}$ means value of potential function at time point $t_y \in T$, and this is the latest available computation of f_p at time point t_x i.e. $t_y = \max_{k \in T} \{k \leq t_x\}$, and similarly, $con^{(t_z)} \in CON$ is an *ECAL* configuration at time point t_z , and this is the latest available computation of $con \in CON$ at time point t_x i.e. $t_z = \max_{k \in T} \{k \leq t_x\}$.

Robot path planning is usually discussed in the context of a configuration space [15] - the notion, which allows the robot to specify the position of any objects in the virtual world. In the presented approach the configuration space is introduced by the cellular automaton *ECAL*.

4 Lazy Potential Function

Using f_p to find the best collision-free route in the dynamically changing environment to the goal requires subsequent calculations $f_p^{(t_1)}, f_p^{(t_2)}, \dots, f_p^{(t_k)}, \dots$

so that at the given time point t_x the latest possible $f_p^{(t_k)}$ is available. Let the frequency of f_p calculation be defined:

$$f_r \stackrel{df}{=} \frac{1}{\max_{t_i \in T_f} |t_{i+1} - t_i|} \tag{5}$$

where $T_f \subset T$ is a set of time points for which f_p is calculated. To be sure that the robot does not overlook changes in the model, the value of f_r should be higher than the frequency of *con* changes. (Then, for every change of *con* there exists at least one f_p calculation, which takes the change into account).

So far, every $f_p^{(t_y)}$ has been calculated for the latest automaton’s configuration $con^{(t_z)}$ i.e. $t_z = \max_{k \in T} \{k \leq t_y\}$. Thus, in the given momentum of time the robot knows the shortest path (in a Manhattan sense) to the goal. Unfortunately, if something large obstructs the path, the values of potential field may change dramatically forcing the robot to look for another way to reach the target. If such changes occur in regular periods of time the robot might run into oscillations, moving simply backward and forward (assuming, of course, that the decision “where to go” is taken on the basis of f_p only). In order to avoid possible oscillation, let $\widehat{f}_p^{(t_x)}$ be defined as:

$$\widehat{f}_p^{(t_{x+1})} = (1 - \xi) \cdot f_p^{(t_{x+1})} + \xi \cdot \widehat{f}_p^{(t_x)} \tag{6}$$

where ξ is a parameter such that $\xi \in [0, 1]$. The higher the value of ξ , the more important the history of the model is. The value of ξ has to be chosen experimentally. It is assumed that initially $\widehat{f}_p^{(t_0)} = f_p^{(t_0)}$. It is worth to noticing that, in this approach the cells with obstacles also need to have some “computable”, i.e. non-infinite, potential values. That is because when they become free, they might also be considered as a path to the goal. In the conducted experiments the potential value assigned to the cells occupied by the obstacles are calculated as follows:

$$f_p(o) = \max_{c \in \eta(o) \setminus O} \{f_p(c)\} + 1 \tag{7}$$

where o is the cell occupied by the obstacle, $\eta(o)$ is its Moore neighborhood, and $O \subset C$ is the set of cells occupied by obstacles. In other words, every obstacle has a potential value locally higher than every non-obstacle cell. An example of value distribution for \widehat{f}_p is shown in Fig. 11b. There are two gates, the upper one with the potential value of 5.8 and the lower one with the potential of 10.3. The difference in potential values stems from the fact, that in the past, the lower gate was closed more frequently than the upper one. Function \widehat{f}_p is called a “lazy” potential function. For every $c \in C$ the value of $\widehat{f}_p(c)$ slowly follows the value of $f_p(c)$. The speed of convergence depends on ξ . The higher ξ is, the more “lazy” is \widehat{f}_p . The lower ξ is, the faster $\widehat{f}_p(c)$ converges to the f_p . For further considerations, it is assumed that the *DWM* and *TDWM* use the lazy version of potential function.

5 Collision-Free Path Finding Algorithm

Using a potential field function to plan the collision-free robot pathway is not a completely new idea. The early potential field based methods come from Khatib [11] and Borenstein and Koren [6]. An implementation of the idea of using a potential field function in robot path planning on cellular automaton was proposed by Behring et al. [5]. Most of the presented methods assume that the environment is static and the once calculated route remains valid until the end of the determined path segment.

In the case of the dynamic environment, the way following the descending gradient is only a suggestion and the robot has to be ready to change its plan when needed. The general scheme of the robot motion algorithm is shown in the listing 1.

```

1 LazyPathPlanner(a)
2 begin
3   start_observe_the_environment()
4   wait_and_learn()
5   :k // goto label
6    $\hat{f}_p \leftarrow \text{get\_current\_}\hat{f}_p()$ 
7   while is_potential_source( $l_a(a)$ )
8     begin
9        $c \leftarrow \text{choose\_next\_cell}()$ ;
10      while is_not_free(c) begin
11        if wait_more() then continue;
12        goto k
13      end
14      move_to(c)
15    end
16 end

```

Listing 1. Collision-free path finding algorithm in a dynamic environment

At the beginning of the *LazyPathPlanner* procedure (listing 1), a separate thread computing subsequent values of potential field (i.e. $\hat{f}_p^{(t_1)}$, $\hat{f}_p^{(t_2)}$, ..., $\hat{f}_p^{(t_k)}$, ...) is started (line: 3). Then, the system must wait a while (line: 4) until the function \hat{f}_p is recalculated a few times by the thread started at (line: 3). After fetching the current potential field distribution (line: 6) the system starts the loop (line: 7) of choosing and moving to (lines: 9 and 14) the subsequent cells. Since the environment is changing dynamically the chosen cell (line: 9) may currently be occupied by the obstacle (i.e. the condition in line 10 is true). In such a case, the robot needs to decide whether to wait until the cell is unblocked (usually waiting a while is not a bad idea) (line: 11), or just get the current \hat{f}_p once again (goto jump from line 12 to line 5) and try to follow a new collision free path.

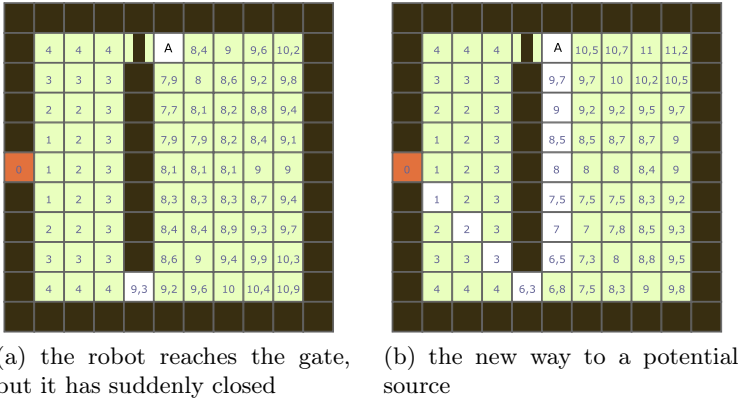


Fig. 2. Lazy potential function - changes in time

To see how the lazy potential function may evolve in time, let us follow the case with two gates, where the upper one opens more often than the lower one (Fig. 1). After reaching the line 6 by the procedure *LazyPathPlanner* the value of \hat{f}_p may be as presented in Fig. 1b. The agent (robot) has planned the shortest path (white cells) leading through the upper gate (mostly open) from its current position to the potential source. Unfortunately, after reaching the gate, it turns out to be closed (Fig. 2a). Thus, the robot starts to wait for one of two situations: the gate opens (probable, since it used to be open) or subsequent updates of \hat{f}_p show the new way to the potential source (Fig. 2b). Then, depending on what happens first, the robot either chooses the upper gate or follows the new path through the lower gate.

6 Simulation Environment

In order to carry out 2D and 3D *DWM* simulations a special middleware library has been written. Thanks to using VRML technology (*Virtual Reality Modeling Language*) this allows efficient modeling 2D and 3D virtual world modling and VRML models designed in third party applications to be imported. Since the library uses Java 3D and Xj3D technology, it is also suitable for making complex, real time visualizations. The library provides a cellular topology, i.e. every object in the virtual world can be interpreted in terms of the cell of the cellular automaton.

Although the library does not validate the laws of physics (an implementation of a simulation algorithm is responsible for that), it supports a simple common sense model checking, for example, if it is possible to check whether two objects do not overlap etc. The two main interfaces of the library: *Agent* and *Sim3dAgents* are simple and easy to use. They contain methods, which allow new agents to be added to the model, the agents to be moved, and additional elements (obstacles) to be added to the model etc. (listing: 2).

```

public interface Agent {
    void loadModel(String fileName);
    void setDelimiters(Vector3f p1,Vector3f p2);
    void setSpeedVector(Vector3f v);
    void setDestination(Vector3f destinationPoint, int time);
    void setDestination(Vector3f destinationPoint,float speed);
    .....
}
public interface Sim3dAgents {
    void createScene(Canvas3D canvas);
    void loadModel(String fileName);
    BranchGroup addGridToScene(int cellNo, int maxDim);
    void setStartViewpoint(Vector3f pos);
    void addAgent(Agent agent);
    boolean removeAgent(Agent agent);
    .....
}

```

Listing 2: DWM middleware visualization library interfaces - excerpt

7 Experiments

The *DWM* with a lazy potential function was implemented on a Java platform and several simulation tests were carried out. For instance, the study case “a barrier with two gates” discussed previously in 2D, was examined in 3D. In the 3D case, the agent is represented by a red helicopter, which aims to reach the window in the wall on the other side of the barrier (Fig. 3).

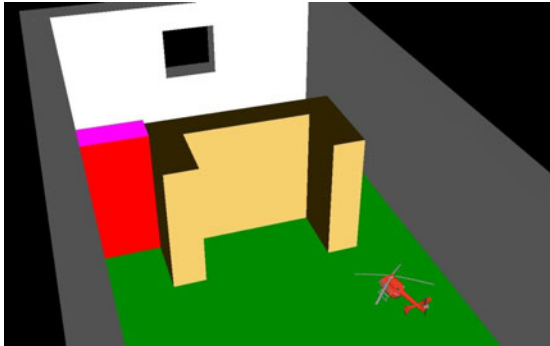


Fig. 3. “Barrier with two gates” test case - 3D view

There are two gates: the red one (on the left side) that opens rarely, and the green one (on the right side), which opens frequently. First, the agent (helicopter) trains the lazy potential function, then takes off and chooses the gateway ensuring greater likelihood of smoothly reaching the goal. The flight trajectory is presented on a 3D cellular space view as a blue path leading from the launch site

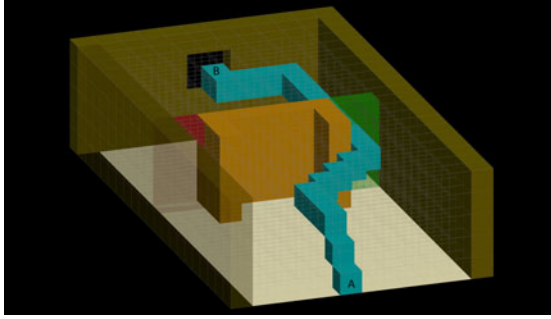


Fig. 4. “Barrier with two gates” test case - 3D cellular space view

to the goal (Fig. 4). In this example, this is the shortest (in terms of potential values assigned to the cells¹) and the most “reliable” way to the target, although not always the case. The “reliable” way may be longer than the shortest one. On the other hand, the average time needed to reach the goal via the “reliable” way strives to be minimal.

An important role in the calculation of the lazy potential function \hat{f}_p is played by ξ . During the conducted experiments, it turned out that $\xi \in [0.7, 0.9]$ ensures the optimal level of “laziness” of \hat{f}_p .

8 Summary and Further Work

In this paper, formal definitions of the *Dynamic World Model* (Eq. 2) and *Temporal Dynamic World Model* (eq. 3) have been presented. Both models have been equipped with the lazy potential function \hat{f}_p gradually changing in time. The function \hat{f}_p has been used to build a collision free path finding algorithm suitable for dynamically changing environments. The usefulness of the algorithm has been confirmed by preliminary tests conducted using a specially constructed library supporting the simulations of 2D and 3D cellular models. The presented solution implicitly assumes that the robot knows the topology and dynamics of its environment. This idealistic assumption may not always be met in practice. Thus, important directions of further research are learning algorithms enabling behavior acquisition from the environment, and pattern recognition algorithms, which allow the class of an object to be recognized, and its further behavior to be estimated. Building more complex environments and testing the presented method on actual robots is also planned.

¹ Due to the clarity of the illustration potential values assigned to the cells are not shown here.

Acknowledgment

This research is supported by AGH University of Science and Technology, contract no.: 11.11.120.859.

References

1. Albus, J.S.: Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics* 21(3), 473–509 (1991)
2. Albus, J.S., et al.: 4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems Version 2.0. Technical report, NIST Interagency (2002)
3. Arkin, R.: Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems* 6(1-2), 105–122 (1990)
4. Arkin, R.: Intelligent Control of Robot Mobility. In: *Handbook of Industrial Robotics*, pp. 317–325. Wiley, Chichester (2007)
5. Behring, C., Bracho, M., Castro, M., Moreno, J.: An algorithm for robot path planning with cellular automata. In: *The Fourth International Conference on Cellular Automata for Research and Industry: Theoretical and Practical Issues on Cellular Automata*, pp. 11–19. Springer, Heidelberg (2000)
6. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots in cluttered environments. *IEEE Transactions on Systems, Man, and Cybernetics* 1, 572–577 (1990)
7. Brooks, R.A.: A robust layered control system for a mobile robot. *IEEE J. Robot. and Auto.* 2(3), 14–23 (1986)
8. Brooks, R.A.: Intelligence without Representation. *Artificial Intelligence* 47, 139–159 (1991)
9. Dudek–Dyduch, E., Waś, J.: Knowledge Representation of Pedestrian Dynamics in Crowd: Formalism of Cellular Automata. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) *ICAISC 2006. LNCS (LNAI)*, vol. 4029, pp. 1101–1110. Springer, Heidelberg (2006)
10. Gat, E.: Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In: *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI 1992*, pp. 809–815. AAAI Press, Menlo Park (1992)
11. Khatib, O.: The potential field approach and operational space formulation in robot control. In: *Adaptive and Learning Systems: Theory and Applications*, pp. 367–377. Plenum Press, New York (1986)
12. Kułakowski, K., Waś, J.: World model for autonomous mobile robot – formal approach. In: *Proceedings of Intelligent Information Systems*, Siedlce, Poland, pp. 37–45 (2010)
13. Lim, G.H., Suh, I.H.: Weighted Action-coupled Semantic Network (wASN) for robot intelligence. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, pp. 2035–2040 (2008)
14. Masehian, E., Sedighzadeh, D.: Classic and heuristic approaches in robot motion planning – a chronological review. *World Academy of Science, Engineering and Technology* 23, 101–106 (2007)
15. Murphy, R.R.: *Introduction to AI Robotics*. MIT Press, Cambridge (2000)
16. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs (1995)
17. Thrun, S.: *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, San Francisco (2003)

VigilAgent for the Development of Agent-Based Multi-robot Surveillance Systems

José M. Gascueña¹, Elena Navarro¹, and Antonio Fernández-Caballero^{1,2}

¹ Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos & Instituto de Investigación en Informática de Albacete (I3A), 02071-Albacete, Spain
{JManuel.Gascuena,Elena.Navarro,Antonio.Fdez}@uclm.es

² Instituto de Desarrollo Industrial (IDI), Computer Vision Research Lab,
c/ Investigación s/n, 02006-Albacete, Spain

Abstract. Usually, surveillance applications are developed following an ad-hoc approach instead of using a methodology to guide stakeholders in achieving quality standards expected from commercial software. To solve this gap, our conjecture is that surveillance applications can be fully developed from their initial design stages by means of agent-based methodologies. Specifically, this paper describes the experience and the results of using a multi-agent systems approach according to the process provided by the *VigilAgent* methodology to develop a mobile robots surveillance application.

Keywords: Multi-agent systems, Agent-based software engineering, Mobile robots, Surveillance.

1 Introduction

At present, it is becoming more and more common to use mobile robots in assisting humans in surveillance tasks. The potential benefits are saving costs, not exposing humans to dangerous situations, and performing surveillance routines more effectively since humans get bored when performing tasks that last long working hours [1]. Sometimes, the capabilities of a single robot are enough to carry out a task (e.g., detecting and following an intruder [4, 9]). However, it is also usual to face scenarios requiring the collaboration among multiple mobile robots to achieve a common goal such as exploring or covering a larger area [13]. These complex tasks can successfully be solved through incorporating multi-agent systems technologies [15] due to the characteristics inherent to agents such as (1) autonomy, necessary to achieve individual and collective goals, and, (2) sociability, indispensable to form robot teams performing cooperatively.

The previous remarks demonstrate the feasibility of using well-known methods, techniques and tools in Agent-Oriented Software Engineering (AOSE) [17] to improve the development of robotic surveillance systems. In fact, agent technology has already been used in several surveillance systems [10]. However, to the best of our knowledge they are not developed following a methodology. Our proposal is to introduce techniques proposed in AOSE to produce well-documented

surveillance applications from requirements to implementation [14], [7], [5]. The exploitation of methodologies in software development provides two main advantages. Firstly, methodologies allow stakeholders to share the same terminology and development process. Second, methodologies are useful to convey more easily the knowledge acquired along a project for the development of future projects showing similar features. The problem in AOSE is that there is not a unique and specific useful methodology without some level of customization [3]. This paper focuses on describing how the process of *VigilAgent* methodology is applied to face the problem of a collection of robots patrolling around a surveillance environment. This methodology has not been developed from scratch as it reuses fragments from Prometheus [11] and INGENIAS [12] methodologies for modelling, and ICARO-T framework [6] for implementation purposes.

The rest of the paper is organized as follows. In section 2, an overview of the phases of the *VigilAgent* methodology is offered. A justification of why Prometheus, INGENIAS and ICARO-T technologies are integrated is provided. Then, section 3 introduces a case study used to demonstrate the applicability of *VigilAgent*. Finally, section 4 offers some conclusions.

2 Overview of the VigilAgent Methodology

The five phases of *VigilAgent* are briefly described next. (1) *System specification* - the analyst identifies the system requirements and the environment of the problem, which are obtained after several meetings arranged with the client; (2) *Architectural design* - the system architect determines what kind of agents the system has and how the interaction between them is; (3) *Detailed design* - the agent designer and application designer collaborate to specify the internal structure of each entity that makes up the system overall architecture produced in the previous phase; (4) *Implementation* - the software developer generates and completes the application code; and (5) *Deployment* - the deployment manager deploys the application according to a specified deployment model.

At this point, several issues about this development process are worth noting. The first one is that the phases named system specification and architectural design in *VigilAgent* coincide with the two first phases of the Prometheus methodology [11]. Another detail is that the third phase of *VigilAgent* (detailed design) uses models of INGENIAS [12]. Finally, notice that code is generated and deployed for ICARO-T framework [6]. Several reasons that are introduced in the following paragraph have conducted to this integration.

Prometheus is significant because of the guidelines proposed to identify which the agents are. Another advantage of Prometheus is the explicit use of the concept *scenario* which is closely related to the specific language used in the surveillance domain. Indeed, a surveillance application is developed to deal with a collection of scenarios. Nevertheless, notice that the last phase of Prometheus has not been integrated in *VigilAgent* because it focuses on BDI agents, and entities obtained during the design are transformed into concepts used in a specific agent-oriented programming language named JACK [16]. This supposes, in

principle, a loss of generality. On the contrary, INGENIAS does facilitate a general process to transform models specified during the design phase into executable code. However, INGENIAS does not offer guidelines to identify the entities of the model; the developer's experience is necessary for their identification. Therefore, *VigilAgent* methodology is not developed from scratch but integrates facilities of both Prometheus and INGENIAS to take advantage of both of them. In [8] the an in-depth comparison between Prometheus and INGENIAS methodologies and supporting tools can be found.

Regarding implementation, the ICARO-T framework has been selected as it provides high level software components that facilitate the development of agent-oriented applications. Moreover, it is independent of the agent architecture; that is, the developer can develop new architectures and incorporate them in the framework. This is a clear difference regarding other agent frameworks such as JACK or JADE [2], which provide a middleware instead of an extensible architecture to establish the communications among agents. An additional advantage are the functionalities already implemented in the framework to automatically carry out component management, application initialization and shutdown, reducing in this way the developers' amount of work and guarantying that the components are under control. These last functionalities are usually not provided by other frameworks.

The novelty of the *VigilAgent* methodological approach is to take advantage of both technology along their development process. Specifically, our contribution in this process consists in integrating the technologies selected (Prometheus, INGENIAS and ICARO-T) by using model transformations. On the one hand, a model transformation to translate Prometheus models into INGENIAS models is proposed (see subsection 3.3). Indeed, both methodologies use a different modeling language. On the other hand, another model transformation is proposed to automatically transform an INGENIAS model into ICARO-T code (see subsection 3.4).

3 Case Study: Collaborative Mobile Robots

The selected case study to illustrate the development of robot-based surveillance applications using the *VigilAgent* methodology process implements the collaboration among several mobile robots to carry out a common surveillance task in an industrial estate. The robots navigate randomly through pre-defined surveillance paths in a simulated environment. When there is an alarm in a building, a robot is assigned the role of the chief, three robots are subordinate to the chief, and the other ones are waiting in rearguard to receive orders from the chief (e.g. to replace a damaged subordinate robot). Failures are discovered by the robot itself when any of its mounted devices (e.g., sonar, laser, camera, etc.) does not work in a right way. The robots perceive that an alarm has occurred through two mechanisms: (1) the security guard notifies robots that an alarm has occurred and where it has taken place, (2) the robot is equipped to perceive an alarm itself when it is close enough to the corner of a building; therefore it does not

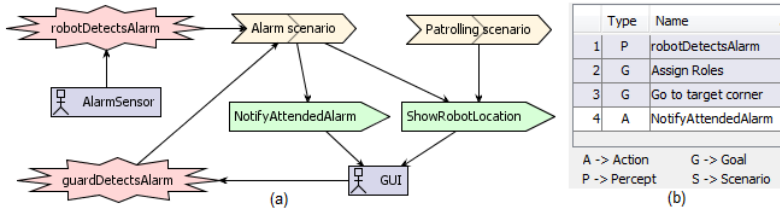


Fig. 1. (a) Analysis Overview Diagram; (b) Steps of the Alarm scenario

have to wait for the security guard announcement. The alarm is covered when a robot coalition (one chief and three subordinates) surrounds the building, that is, the robots that form a coalition are located at the four corners of the building where the alarm has occurred. In this case study three hypotheses are assumed: (1) several alarms do not take place simultaneously, (2) robots do not collide as the streets are wide enough, and, (3) robots navigate from corner to corner.

3.1 System Specification

The analyst starts the system specification phase by developing an *analysis overview diagram*, which shows the interactions between the system and the environment (see Fig. 1a). At this level, firstly, an *AlarmSensor* actor for the device mounted on the robot to detect alarms has been identified. There is also a *GUI* actor representing the user interface that supports the human interaction with the system, that is, it shows the monitoring activity to the security guard, and the commands that he/she can send to the system to announce an alarm, to simulate the failure of a robot, and so on. On the one hand, the information that comes from the environment is identified as *percepts*. For example, the command issued by the guard in order to notify to the robots that an alarm has been triggered (*guardDetectsAlarm*) and the signal captured automatically by the robot device when it is close to a building with an alarm (*robotDetectsAlarm*). On the other hand, every operation performed by the system on the actors is identified as an *action*. For example, an the alert message displayed on the user interface to notify that an alarm has been attended (*NotifyAttendedAlarm*). Finally, relations with the scenarios identified to navigate along the environment when there is no alarm and when there is an alarm are established (see *Patrolling* and *Alarm* scenarios, respectively).

A scenario is a sequence of structured steps - labelled as action (A), percept (P), goal (G), or other scenario (S) - that represents a possible execution pathway of the system. As an example, Fig. 1b illustrates the process performed by the system to attend an alarm. This scenario begins when an alarm has occurred and it is perceived using the sensor mounted on the robot (step 1). Alternatively, the scenario also starts with *guardDetectsalarm* percept. Then, the process to assign roles for each robot to deal with the alarm is started (step 2). Afterwards, robots playing roles chief and subordinates go to the assigned target corners (step 3).

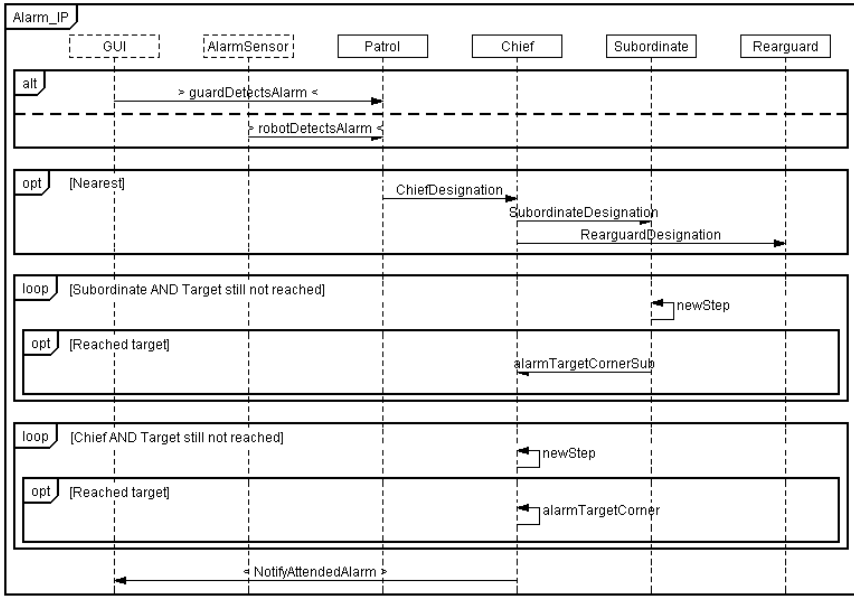


Fig. 2. Alarm interaction protocol

The scenario finishes showing a message when four robots are surrounding the building with alarm (step 4).

Notice that a scenario has a goal representing the objective to be achieved by the scenario. This goal is decomposed into new sub-goals to denote how to achieve the parent goal. For instance, the general goal *Alarm* associated to *Alarm scenario* has been refined into two goals (*Assign Roles* and *Go to target corner*).

3.2 Architectural Design

A relevant task in architectural design phase is to define the conversation entities (interaction protocols, IP) for describing what should happen to realize the specified goals and scenarios. In this case study, IP are used to graphically represent (i) interactions among robots, and, (ii) interactions between a robot and the environment. For example, Fig. 2 details the *Alarm_IP* interaction protocol internal structure. As can be noticed, it involves the roles that four robots and two actors can play to deal with alarm situations (identified by the dotted squares in the diagram). Firstly, the robot is patrolling and captures a percept sent by the *GUI* or *AlarmSernsor* actor when the alarm is triggered. This perception contains the identifier of the building with alarm. Secondly, if a robot is the closest to the building then it sends itself a *ChiefDesignation* message to become the chief robot. Then, on the one hand, the chief sends a *SubordinateDesignation* message to the three next closer robots to convert them into subordinate robots and to notify the target corner where they must go. On the other hand, it sends a *RearguardDesignation* message to the rest of the robots. From this moment

on, each subordinate robot is continuously sending itself a *newStep* message to move towards its target corner until the assigned target has been reached; this is communicated to the chief robot sending an *alarmTargetCornerSub* message. A similar approach is used by the chief to go the chief target corner. Finally, the chief displays a text message on the GUI to communicate that the alarm has been attended (four robots surround the building). Moreover, two protocols are specified to describe the management of failures and patrolling performed for the robots when there is no alarm, respectively.

Finally, another task performed during the architectural design phase is to identify the information managed by the agents or the beliefs describing agent knowledge about the environment or itself. For example, *RobotLocation* data is used to store the robots location, whereas *Environment* data provides information about the simulated environment (industrial estate dimensions, building where the alarm takes place, robots that do not work well, and robots initial locations). Let us point out that this data is public for all agent instances. So, any agent (robot) knows the location of the other ones and the information about the environment.

3.3 Detailed Design

The *VigilAgent* models of the detailed design phase are developed using INGENIAS concepts, which are different from the concepts used in the previous phases. Therefore, transformations from Prometheus to INGENIAS models are carried out to be able to perform the third modelling phase (detailed design). Four conceptual mappings have been developed [8] to transform the structures that involve percepts, actions, messages and data related to agents. These mappings have been inferred considering both the definition of these concepts, and how each Prometheus structure can be modelled using an INGENIAS equivalent structure.

For example, a *percept* is a piece of information from the environment received by means of a sensor. *Percepts* are sent by *actors* (Actor \rightarrow Percept) and received by agents (Percept \rightarrow Agent). In INGENIAS, any software or hardware that interacts with the system and that can not be designed as an agent is considered an *application*; and every agent that perceives changes in the environment must be in the *environment model* associated to an application. Therefore, as Fig. 3 shows (arrow 1), the percepts of a Prometheus agent can be triggered in INGENIAS by specifying a collection of *operations* in an *application*. A Prometheus percept has a field, *Information carried*, to specify the information it carries. As Fig. 3 depicts (arrow 2), in INGENIAS this information is described in a type of event named *ApplicationEventSlots* that is associated to the *EPerceives* relation established between the agent and the corresponding application. Notice that the Prometheus agent and actor concepts have been directly mapped to INGENIAS agent and application concepts (see arrows 3 and 4, respectively).

Another interesting example is how Prometheus data are translated into equivalent INGENIAS concepts. It is important to mention that Prometheus data are of any granularity. So, any information represented with a simple data

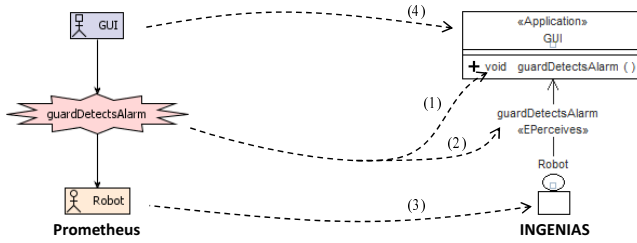


Fig. 3. Mapping information related with Prometheus percepts into INGENIAS

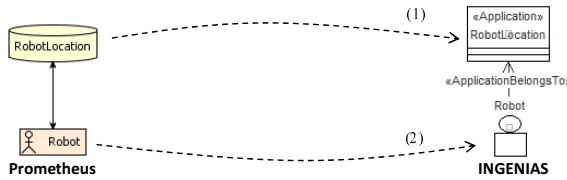


Fig. 4. Mapping information related with Prometheus data into INGENIAS

type (e.g. a string or boolean) or a complex data type (e.g. database) is a data. The model fragment depicted on the left in Fig. 4 represents that the *Robot* agent writes and reads information (*RobotLocation* data) about the robot's location. The relations $\text{Agent} \rightarrow \text{Data}$ and $\text{Data} \rightarrow \text{Agent}$ express in Prometheus that data is written and read by the agent, respectively. These structures are specified in the INGENIAS environment model with the *Agent - Application-BelongsTo* \rightarrow *Application* structure, when data has coarse granularity (e.g. a database or a complex data structure to store non-persistent information). Notice that the application is not the data, but the entity that provides methods to data management. Regarding simple data, they are not translated to an INGENIAS concept but they are declared when the implementation is carried out. Finally, let us highlight that action and message concepts in Prometheus are equivalents in INGENIAS to operation defined in the applications, and interaction unit concepts, respectively.

Once the transformation has been performed, new activities should be carried out to complete the modelling. Firstly, it is necessary to identify the task performed by agents for every received perception or message. After that, the behaviour of each agent is specified using information about tasks, received percepts and messages. The behaviour is modelled with a finite state automaton, where the states represent concrete situations of the agent life cycle. For example, the state diagram interpretation corresponding to the reactive agent that controls a robot is as follows (see Fig. 5) - the term event is used to refer either to a perception or a message.

- There are three kinds of states: initial (*InitialState*), final (*FinalState*) and intermediate (e.g., *AlarmDetection*, *Rearguard*, and so on).

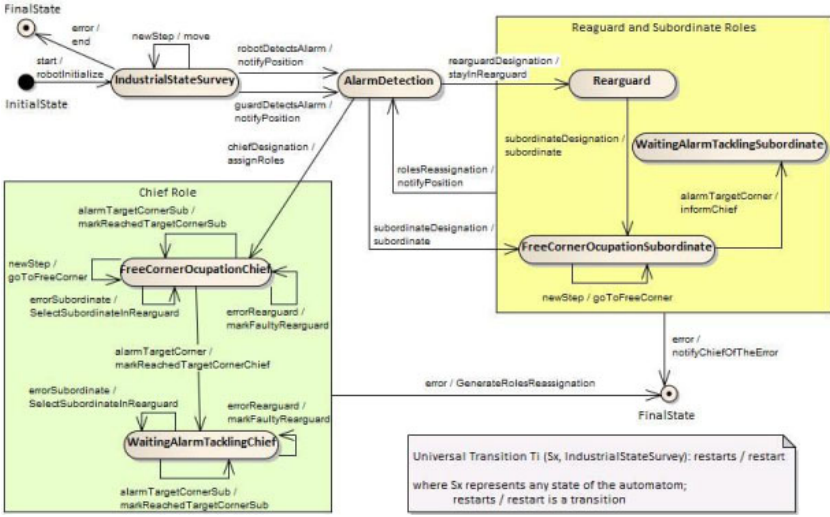


Fig. 5. State diagram for the robot's behaviour control

- The agent starts executing when it receives a start event, causing the agent to change to *IndustrialStateSurvey* state and to execute the *robotInitialize* action. The agent does not consult if there are events associated with transitions that go from *IndustrialStateSurvey* state to *robotInitialize*. Once the agent has been initialized, it navigates randomly (cycling in *IndustrialStateSurvey* state) until an alarm appears (a *robotDetectsAlarm* or *guardDetectsAlarm* event arrives). In this case, *notifyPosition* action is executed to determine who the chief is. After that, *assignRoles* action produces the assignment of subordinate and rearguard roles to the other agents. The agent with chief role enters the states enclosed in the 'Chief Role' boundary, whereas the agents with other roles go to states enclosed into 'Rearguard and Subordinate Roles' boundary.
- There is a kind of particular transition, the universal transition, which is valid for any automaton state. This transition takes place for a given input; the action is executed and the automaton transits to the next state, regardless of the automaton's state.
- In *robotInitialize* action the agent sends itself the *newStep* event to start moving the robot. *End* action marks the agent as damaged. *Restart* action restarts the simulation process, and, afterwards the agent sends itself a *newStep* event. The rest of actions are described in Table 1, where the role played by the agent is mentioned only when it is meaningful.

In Fig. 5 the following notation is adopted for the purpose of clarity. A transition that goes from a boundary to a state means that such a transition is able to go from any state enclosed into the boundary to the specified target state (see, for instance, *rolesReassignment / notifyPosition*).

Table 1. Description of the automaton's actions

Action	Description
move	The agent sends itself a <i>newStep</i> event if there is no alarm.
notifyPosition	Determine if the agent becomes the chief. The chief is the agent closer to the alarm and the ties are solved in favour of the agent that has a lower index. The agent sends itself a <i>ChiefDesignation</i> event if it becomes the chief.
stayInRearguard	The agent updates its role as rearguard and learns who the chief is. The robot does not move while it plays this role.
subordinate	The agent (1) updates its role as subordinate and learns who the chief is, and, (2) sends itself a <i>newStep</i> event.
assignRoles	The agent (1) updates its role as chief, (2) assigns to what corner the chief should go, (3) assigns to what corners the three next closest agents to the alarm should go to, and sends them a <i>subordinateDesignation</i> event that contains the following information: the target corner that it should occupy and who the chief robot is, (4) send to the other agents a <i>rearguardDesignation</i> event, and finally, (5) sends itself a <i>newStep</i> event. The <i>notifyPosition</i> action may be consulted to know how the ties are solved.
goToFreeCorner	If the chief/subordinate agent is on the assigned target corner, then it sends itself an <i>alarmTargetCorner</i> event; otherwise it determines to what corner it moves next, and it sends itself a <i>newStep</i> event.
notifyChiefOf-TheError	If the agent is a rearguard, then it sends an <i>errorRearguard</i> event to the chief; if it is a subordinate agent, then it sends an <i>errorSubordinate</i> event, which contains its identification number to the chief. In both cases, the agent marks the controlled robot as damaged.
informChief	The subordinate agent sends to the chief agent an <i>alarmTargetCornerSub</i> event, which contains the subordinate agent identification number.
markReached-TargetCornerSub	The chief agent (1) marks the target corner that the subordinate agent has occupied, (2) increases the number of occupied corners, and, (3) notifies the user when four target corners have been occupied.
markReached-TargetCornerChief	The chief agent (1) marks the target corner that it has occupied, (2) increases the number of occupied corners, and, (3) notifies the user when four target corners have been occupied.
selectSubordinate-InRearguard	The chief agent (1) increases the number of damaged robots, (2) identifies the closest rearguard agent to the target corner to be occupied by the damaged subordinate agent, and, (3) sends a <i>subordinateDesignation</i> event that contains the target corner and the chief identification number.
markFaulty-Rearguard	The chief agent (1) marks the rearguard agent that sent an <i>errorRearguard</i> event as damaged, and, (2) increases the number of damaged robots.
generateRoles-Reassignment	The chief agent (1) marks itself as damaged, (2) increases the number of damaged robots, and, (3) sends to the rest of agents a <i>rolesReassignment</i> event that contains the location of the building where the alarm occurred.

3.4 Implementation and Deployment

Now, we are in front of the second major contribution of the *VigilAgent* methodology. Our approach considers that the tool supporting INGENIAS, Ingenias Development Kit (IDK) [12], is an exceptional agent tool to develop a Model-To-Text transformation for generating code for any target language chosen. This is ICARO-T in *VigilAgent*, as it provides the necessary functionalities for developing new modules capable to carry out this task. These modules are developed following a general process based on both the definition of specific templates for each target platform, and procedures to extract information from INGENIAS models.

To facilitate the implementation and deployment, we have developed modules to automatically generate ICARO-T code of the system being developed from the models specified with IDK. The process for using our modules is carried out as follows. (1) The *INGENIAS ICARO-T Framework generator module (IIF)* is used to automatically generate code for the detailed design specification. *IIF* generates several XML files that describe the behaviour of each agent, java classes for each agent and application, and the XML file describing the application deployment. (2) The developer manually inserts code in the protected regions of the generated java classes and implements those new classes he/she needs. (3) The developer uses the *ICAROTCodeUploader* module to update the model with the modifications introduced in the protected regions. Finally, the script file generated by *IIF* module is executed by the deployment manager to launch the developed application.

4 Conclusions

The development of an application using *VigilAgent* has been described in this paper. The learning curve of *VigilAgent* can be steep at first because users must get used to different terms that have the same meaning depending on the technology used in each phase (Prometheus and INGENIAS for modelling, and ICARO-T for implementation). However, this disadvantage is overcome thanks to the two transformations that are automatically executed.

It is worth pointing out that the time spent learning how to develop and implement the *INGENIAS ICARO-T Framework generator* and the *ICAROT-CodeUploader* modules described in section 3.4 was two months and fifteen days. This effort is rewarded as new applications are modelled and implemented. Our future work consists in applying *VigilAgent* methodology to new case studies and domains.

Acknowledgements

This work was partially supported by Spanish Ministerio de Ciencia e Innovación under TIN2010-20845-C03-01 and CENIT A-78423480 grants, and by Junta de Comunidades de Castilla-La Mancha under PII2I09-0069-0994 and PEII09-0054-9581 grants.

References

1. Anisi, D.A., Thunberg, J.: Survey of patrolling algorithms for surveillance UGVs. Scientific report (2007), www.math.kth.se/~anisi/articles/CTAPP_survey.pdf
2. Bellifemine, F., Caire, G., Greenwood, D.: *Developing Multi-Agent Systems with JADE*. John Wiley and Sons, Chichester (2007)
3. Cossentino, M., Gaglio, S., Garro, A., Seidita, V.: Method fragments for agent design methodologies: From standardisation to research. *International Journal of Agent-Oriented Software Engineering* 1(1), 91–121 (2007)
4. Fernández-Caballero, A., Castillo, J.C., Martínez-Cantos, J., Martínez-Tomas, R.: Optical flow or image subtraction in human detection from infrared camera on mobile robot. *Robotics and Autonomous Systems* 58(12), 1273–1280 (2010)
5. Fernández-Caballero, A., Gascueña, J.M.: Developing multi-agent systems through integrating Prometheus, INGENIAS and ICARO-T. *Communications in Computer and Information Science (Agents and Artificial Intelligence)* 67, 219–232 (2010)
6. Garijo, F.J., Polo, F., Spina, D., Rodríguez, C.: *ICARO-T User Manual*. Technical Report, Telefonica I+D (2008)
7. Gascueña, J.M., Fernández-Caballero, A.: Agent-based modeling of a mobile robot to detect and follow humans. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009*. LNCS, vol. 5559, pp. 80–89. Springer, Heidelberg (2009)
8. Gascueña, J.M., Fernández-Caballero, A.: Prometheus and INGENIAS agent methodologies: A complementary approach. In: Luck, M., Gomez-Sanz, J.J. (eds.) *AOSE 2008*. LNCS, vol. 5386, pp. 131–144. Springer, Heidelberg (2009)
9. Gascueña, J.M., Fernández-Caballero, A.: Agent-oriented modeling and development of a person-following mobile robot. *Expert Systems with Applications* 38(4), 4280–4290 (2011)
10. Gascueña, J.M., Fernández-Caballero, A.: On the use of agent technology in intelligent, multi-sensory and distributed surveillance. *The Knowledge Engineering Review* (2011) (in press)
11. Padgham, L., Winikoff, M.: *Developing Intelligent Agents Systems: A Practical Guide*. John Wiley and Sons, Chichester (2004)
12. Pavón, J., Gómez-Sanz, J., Fuentes, R.: The INGENIAS Methodology and Tools. In: *Agent-Oriented Methodologies*, pp. 236–276. Idea Group Publishing, USA (2005)
13. Rogge, J.A., Aeyels, D.: A strategy for exploration with a multi-robot system. In: *Informatics in Control, Automation and Robotics. Lecture Notes in Electrical Engineering*, vol. 24, part II, pp. 195–206 (2009)
14. Sokolova, M.V., Fernández-Caballero, A.: Facilitating MAS Complete Life Cycle through the Protégé-Prometheus Approach. In: Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2008*. LNCS (LNAI), vol. 4953, pp. 63–72. Springer, Heidelberg (2008)
15. Weiss, G.: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge (1999)
16. Winikoff, M.: *Jack Intelligent Agents: An Industrial Strength Platform. Multi-Agent Programming Languages, Platforms Applications*, pp. 175–193 (2005)
17. Wooldridge, M.: Agent-based software engineering. *IEEE Proceedings - Software Engineering* 144(1), 26–37 (1997)

Searching Targets Using Mobile Agents in a Large Scale Multi-robot Environment

Tsukasa Abe¹, Munehiro Takimoto¹, and Yasushi Kambayashi³

¹ Department of Information Sciences, Tokyo University of Science, Japan

² Department of Computer and Information Engineering,
Nippon Institute of Technology, Japan

Abstract. This paper presents a framework for controlling multiple robots connected by communication networks. Instead of making multiple robots pursue several tasks simultaneously, the framework makes mobile software agents migrate from one robot to another to perform the tasks. Since mobile software agents can migrate to arbitrary robots by wireless communication networks, they can find the most suitably equipped and/or the most suitably located robots to perform their task. In the previous papers, we have shown that this manner of controlling multiple robots can decrease the number of required robot resources in the three-robot case. The three robots demonstrated that they could suppress the energy consumption through a mobile agent based approach. In this paper, we pursue a case of large scale multiple robots. We have implemented a simulator for a large number of robots based on our framework, and have demonstrated the efficiency and the scalability. We report the observations we found in the numerical experiments.

Keywords: Mobile agent, Dynamic software composition, Intelligent robot control.

1 Introduction

In the last decade, robot systems have made rapid progress not only in their behaviors but also in the way they are controlled. In particular, a control system based on multiple software agents can control robots efficiently. Multi-agent systems introduced modularity, reconfigurability and extensibility to control systems, which had been traditionally monolithic. It has made easier the development of control systems on distributed environments such as multi-robot systems.

On the other hand, excessive interactions among agents in the multi-agent system may cause problems in the multiple robot environments. In order to mitigate the problems of excessive communication, mobile agent methodologies have been developed for distributed environments. In a mobile agent system, each agent can actively migrate from one site to another site. Since a mobile agent can bring the necessary functionalities with it and perform its tasks autonomously, it can reduce the necessity for interaction with other sites. In the minimal case,

a mobile agent requires that the connection is established only when it performs migration [1].

The model of our system is a set of cooperative multiple mobile agents executing tasks by controlling a pool of multiple robots [2]. The property of inter-robot movement of the mobile agent contributes to the flexible and efficient use of the robot resources. A mobile agent can migrate to the robot that is most conveniently located to a given task, e.g. closest robot to a physical object such as a soccer ball. Since the agent migration is much easier than the robot motion, the agent migration contributes to saving power consumption. Here, notice that any agents on a robot can be killed as soon as they finish their tasks. If the agent has a policy of choosing idle robots rather than busy ones in addition to the power-saving effect, it would result in more efficient use of robot resources.

We have proposed our model in the previous paper [3,4], and have also shown the effectiveness of saving power consumption and the efficiency of our system in a case of three physical robots [4]. In this paper, we focus our attention on the aggregate behaviors of a large number of robots based on our framework in order to investigate properties of our framework for practical use. We show the results through the numerical experiments on a simulator.

The structure of the balance of this paper is as follows. In the second section we describe the background. The third section describes the mobile agent class library that we have developed for controlling multiple robots. In our robot control system, the mobility that the software mobile agent system provides is the key feature that supports the ability of adding new functionalities to intelligent robots in action. The fourth section shows an example of intelligent robot systems in which robots search multiple target objects cooperatively. In the fifth section, we demonstrate how the properties of mobile agents can contribute to efficient use of robot resources through numerical experiments based on a simulator. Finally, we conclude in the sixth section.

2 Background

The traditional structure for the construction of intelligent robots is to make large, often monolithic, artificial intelligence software systems. The ALVINN autonomous driving system is one of the most successful of such developments [5]. Putting intelligence into robots is, however, not an easy task. An intelligent robot that is able to work in the real world needs a large-scale knowledge base. The ALVINN system employs neural networks to acquire the knowledge semi-automatically [6]. One of the limitations of neural networks is that it assumes that the system is used in the same environment as that in which it was trained. Indeed, many intelligent robots lack a mechanism to adapt to a previously unknown environment.

On the other hand, multi-agent robotic systems are recently becoming popular in RoboCup or MIROSOT [7]. In traditional multi-agent systems, robots communicate with each other to achieve cooperative behaviors. The ALLIANCE architecture, developed in Oak Ridge National Laboratory, showed that cooperative

intelligent systems could be achieved [8]. The architecture is, however, mainly designed to support self-adaptability. The robots in the system are expected to behave without external interference, and they show some intelligent behaviors. The observed intelligence, however, is limited due to the simple mechanism called *motivation*. Robots' behaviors are regulated by only two rules *robot impatience* and *robot acquiescence*. These rules are initially defined and do not evolve. In contrast, the goal of our system is to introduce intelligence and knowledge into the robots after they start to work [2]. Therefore, our system does not have any learning mechanism or knowledge acquiring mechanism. All the necessary knowledge is sent as mobile agents from other robots or the host computer.

3 Mobile Agent Controlling Robots

We assume that a mobile agent system consists of mobile agents and places. Places provide runtime environments, through which mobile agents achieve migration from one environment to other environments. When a mobile agent migrates to another place, not only the program code of the agent but also the state of the agent can be transferred to the destination. Once an agent arrives at another place through migration, it can communicate with other mobile agents on that place.

The mobile agent system we have used to control robots is based on an existing mobile agent system, called AgentSpace, developed by I. Satoh [9]. AgentSpace provides the basic framework for mobile agents. It is built on the Java virtual machine, and agents are supposed to be programmed in Java language. We have extended AgentSpace and developed an agent library for controlling mobile robots.

We have implemented an agent library *Robot* as an extension of AgentSpace that includes methods for controlling robots. In order to implement the methods, we have taken advantage of primitives of ERSP. ERSP is a software development kit with high-level interfaces tailored for controlling robots. These interfaces provide several high-level means for control such as driving wheels, detecting objects through a camera, checking obstacles through supersonic sensors, and intelligent navigations. They are written in C++, while mobile agents are described as Java classes that extend `Agent` class of AgentSpace. Therefore, we have designed *Robot* library that uses these interfaces through JNI (Java Native Interface). The library *Robot* has interfaces that are supposed to be implemented for the following methods:

initialize initializes flags for inputs from a camera and sensors,
walk makes a robot move straight within required distance,
turn makes a robot turn within required degree,
setObjectEvent resets the flag for object recognition with a camera,
setObstacleEvent resets the flag for supersonic sensors,
getObject checks the flag for object recognition,
getObstacle checks the flag for the sensors, and
terminate halts all the behaviors of a robot.

4 Robot Controller Agents for Target Searcher

In this section, we demonstrate that our model, which is a set of cooperative multiple mobile agents, is an efficient way to control multiple robots. In our robot control system, each mobile agent plays a role in the software that controls one robot, and is responsible to complete its own task. One agent with one specific task migrates to one specific robot to perform that task. In this manner, an agent can achieve several tasks one by one through the migration. This scheme provides more idle time for each robot, and allows other agents to use the idle robots for incomplete tasks. In that way, this scheme contributes in decreasing the total time of completing all the tasks. We will show these advantages in the numerical experiments.

4.1 Controlling Robots

An intelligent multi-robot system is expected to work in a distributed environment where communication is relatively unstable and therefore where fully remote control is hard to achieve. Also we cannot expect that we know everything in the environment beforehand. Therefore intelligent robot control software needs to have the following features: 1) It should be autonomous to some extent. 2) It should be extensible to accommodate the working environment. 3) It should be replaceable while in action. Our mobile agents satisfy all these functional requirements.

In order to construct a realistic simulator, we have extracted information by observing the behaviors of three wheeled mobile robots, called PIONEER 3-DX. Each robot has two servo-motors with wheels, one camera and sixteen sonic sensors. The power is supplied by a rechargeable battery. Each robot holds one notebook computer as its host computer. Our control agents migrate to these host computers by wireless LAN.

4.2 Searching a Target

Let us consider how to program a team of multiple robots to find a target. For such a task, the most straightforward solution would be to make all robots search for the target simultaneously. If the targets were comparatively fewer than the robots, however, most robots would move around in vain, consuming power without finding anything.

This problem can be more serious in our model where any robots can be shared by any agents, because the robots to which an agent with a new task is going to migrate may be already occupied by another agent with some different task. Especially, consider a case where the robots are working in an area where communications on wireless LAN are difficult. In such a case, even if one of the working robots finds the target, the other robot may not be able to know that fact. As a result, most robots continue to work to search that target in vain until time-out. Thus, this searching strategy could not only wastes the power but also increase the total costs of the multiple robots in aggregate. On the other hand,

our strategy of using mobile agents achieves the suppression of the total due to the efficient use of idle resources as well as saving power consumption.

The core of our idea is finding the nearest robot to the target by using agent migration. Initially, an agent is dispatched from the host machine to a nearby robot in the multi-robots system. Then, the agent hops among the robots one by one and checks the robot's vision in order to locate the target until it reaches the robot that is closest to the target. Until this stage, robots in the multi-robot system do not move; only the mobile agent migrates around so that robots can save power.

Once the agent finds the target, it migrates to the closest robot and makes the robot move toward the target. In our strategy, since only one robot dedicates to a particular task at a time, it is essentially similar to making each robot special for each task. Since the migration time is negligible compared to robot motion, our strategy is more efficient than such as we described before. If the agent visits all the robots without finding the target, the agent makes the last one move around randomly with wheels in order to find the target.

In our current multi-robot system, the robots' vision does not cover 360 degrees. Therefore a robot has to rotate to check its circumstance. Rotating a robot at its current position may capture the target and another robot closer to the target. Then the agent migrates to the more conveniently located robot. Take note that the rotation requires much less movement of the wheels than exploration, and it contributes to the power saving.

Details of our searching algorithm are the followings: 1) an agent chooses an arbitrary robot to which it migrates, and performs the migration, 2) as the agent arrives on the robot, it makes that robot rotate, where if the robot to which the agent migrates has been occupied by another agent, it migrates to another robot further, 3) if the target is found, the agent makes the robot move to that direction; otherwise, goes back to step 1, and 4) at this stage, if all robots have been tried without finding the target, the agent makes the last robot do random-walk until it can find a target.

We have implemented this algorithm as an AgentSpace agent *search*. As soon as *search* agent has migrated to a robot, its *arrive()* method is invoked. *Arrive()* checks whether there are any other agents on the current robot or not. That can be achieved by checking the number of agents' id's. This is achieved by calling *context.getAgents()*. If it finds only its own agent id, it means that there are no other agents. If it finds no other agents, it invokes *behavior()* as the main behavior of the robot. Otherwise, it immediately migrates to another robot in order to avoid interferences with the other agents.

The method *behavior()* first makes the robot rotate within 360 degrees to look around its circumstance. If it finds something, it stops the rotation of the robot, and sets the flag that indicates it detects an object. At this time, what is found can be checked through *Robot.getObject()*. As a result, if the return value corresponds to "TARGET1", it makes the robot go straight through *Robot.walk()*. Otherwise, it checks whether it has visited all the robots through *isExhausted()*. If there is no more robots to visit, it invokes *randomWalk()*, and makes the

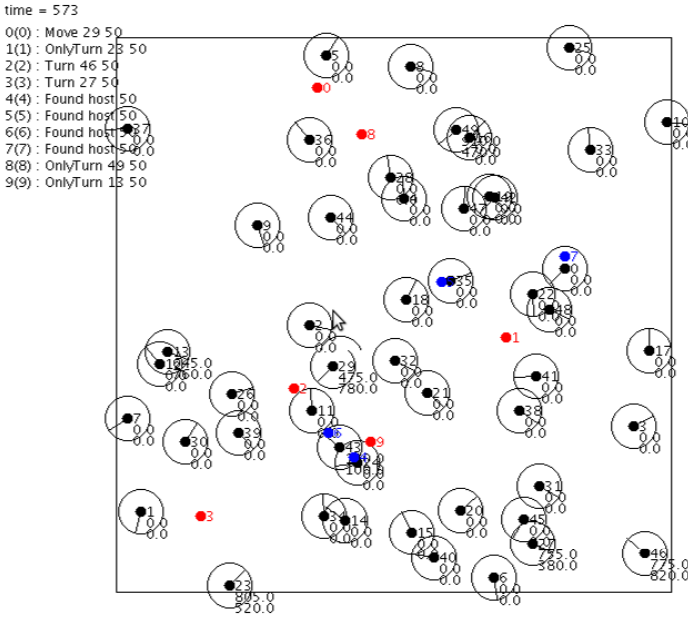


Fig. 1. A snapshot of a running simulator

robot walk randomly in order to find the target at different angles. Otherwise, it migrates to one of the other robots.

5 Experimental Results

In order to demonstrate the effectiveness of our system in a realistic environment, we have implemented a simulator of the target search based on the real multi-robot system described in the previous section, and have conducted numerical experiments on it. On the simulator, moving and rotating speed of robots, and lags required in agent migration and object recognition are based on real values in the previous experiments using PIONEER 3-DX with ERSP [4]. In the experiments, we set a condition where fifty robots are scattered in a 500×500 square field in the simulator, where searching each target corresponds to a distinct task. We have compared our approach based on mobile agents with other two strategies, AllForEach and EachForEach explained respectively as follows:

- AllForEach:** makes all robots move around for each target one by one, and
- EachForEach:** allocates specific targets to each robot. Each robot searches its own targets, and does not do any other tasks.

We have recorded the total moving distance and the total time of the robots that perform all the strategies. We have evaluated the results by changing the two

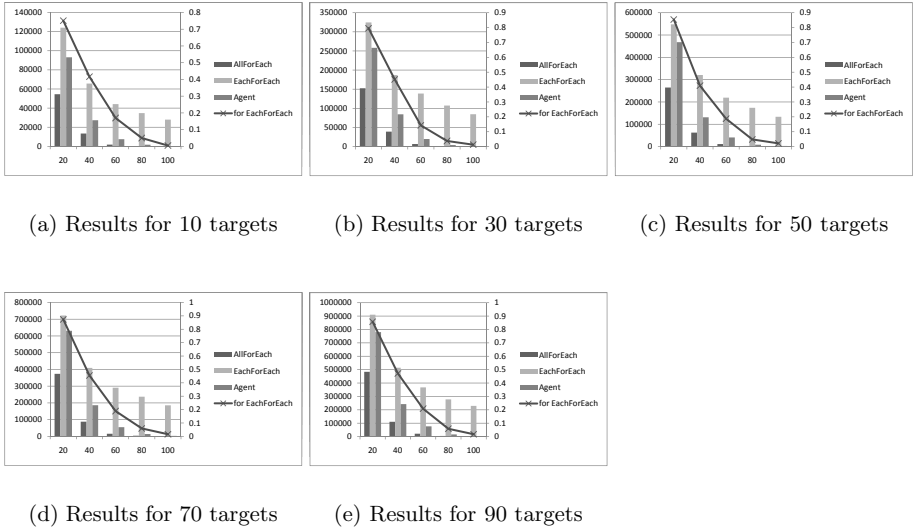


Fig. 2. The total moving distances for the viewFor width: 20, 40, 60, 80, and 100

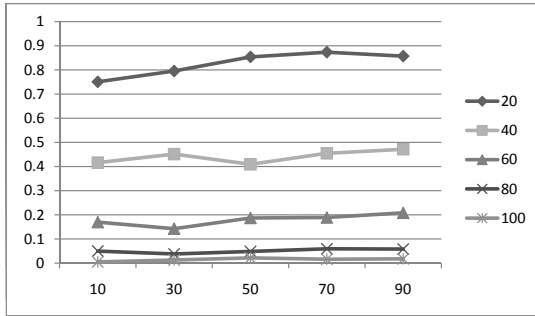


Fig. 3. The total moving distance over the numbers of targets

parameters. They are the number of targets and the width of a view. The view means a circle with a robot as a center, and the robot can detect any objects in the circle as shown in Fig. 1. It is reasonable to assume that energy consumption of servomotors is linear to moving distance.

Bar charts of Fig. 2(a)–(d) show each of the total moving distances for 10, 30, 50, 70, and 90 targets respectively. The AllForEach strategy seems to achieve the least energy consumption over any number of targets. In some cases, it shows itself as the most efficient method for searching multiple targets. This approach, however, may consume a lot of energy when the condition of connections among robots is intermittent. Even though one robot finds a target, other robots may

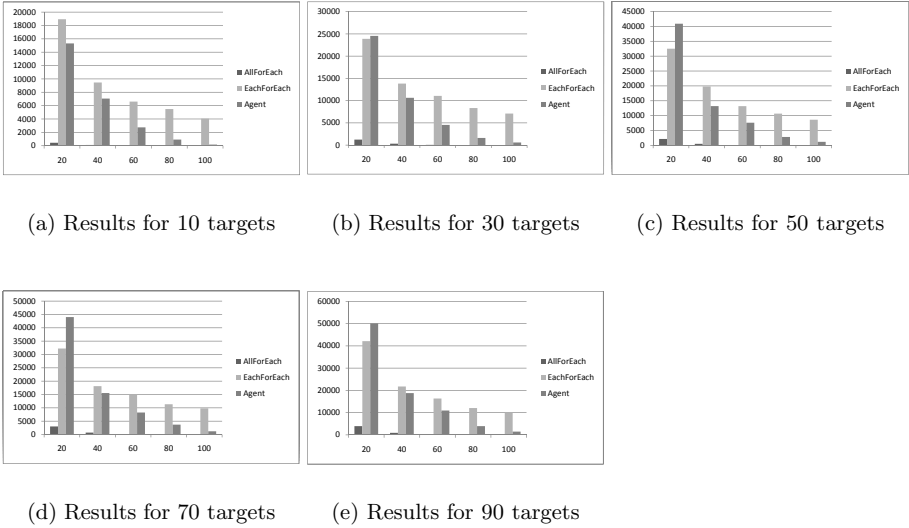


Fig. 4. The total time for each the view width: 20, 40, 60, 80, and 100

not be able to know the fact. In that case, the task for searching one target might consume all the allowances for the entire team of robots. In our mobile agent based approach, on the other hand, a mobile agent is fixed on a certain robot after its migration, and as soon as the robot achieves the task, the entire multi-robot system turns to pursue the next object. As a result, the behavior of entire multi-robot system becomes effectively similar to the EachForEach strategy but it performs the same task just more efficiently.

The agent system displays a remarkable saving of energy consumption compared to the EachForEach strategy. Furthermore, the more the width of a view increases, the more efficiency Agent gains than EachForEach, as shown by the line chart representing the ratio of the Agent to the EachForEach in Fig. 2(a)–(d).

Meanwhile, Fig. 3 shows the ratio of the total moving distance of the Agent strategy to the EachForEach strategy for each view width over the various numbers of targets. The flat lines illustrate the constant advantage of the Agent strategy over the EachForEach strategy regardless of the number of targets.

Fig. 4(a)–(d) shows the total time for searching out all the 10, 30, 50, 70, and 90 targets respectively. Since the total duration time is proportional to the total moving distance, the AllForEach strategy seems to be the most efficient among the three strategies. But it is not practical as mentioned above. On the other hand, the Agent strategy makes mobile agents occupy robot resources not so long as the other approaches do, and produces more idle resources. Then other new agents with other tasks can effectively use the idle resources through migration. We, however, observe that the Agent strategy shows less efficiency than that of the EachForEach where the width of a view is 20 shown in Fig. 4(b)–(d). In such cases, the Agent strategy often fails to find any target during the migration step

due to the restricted view, and causes robots to randomly walk. We can conclude that the wider the view becomes, the more efficiently Agent works.

Thus, we believe that our approach is practical enough for controlling agent based multi-robots system in the real world in terms of the total cost and energy consumption.

6 Conclusions

We have presented a novel framework for controlling intelligent multi-robots. The framework helps users to construct intelligent robot control software by migration of mobile agents. Since the migrating agents can dynamically change the functionalities of the robot which they control, the control software can be flexibly assembled while they are running. Such a dynamically extending the control software by the migration of mobile agents enables us to make the base control software relatively simple, and to add functionalities one by one as we know the working environment. Thus we do not have to make the intelligent robot smart from the beginning or make the robot learn by itself. We can send intelligence later as new agents.

We have implemented a simulator that simulates the behaviors of a large scale team of cooperative search robots to show the effectiveness of our framework, and demonstrated that our framework contributes to suppressing the total cost of a multi-robot system. The numerical experiments show the volume of saved energy is significant. They demonstrate the superiority of our approach over more traditional non-agent based approaches.

Our future directions for research will include the addition of security features, refinement of the implementation of dynamic extension, additional proof of concept for dynamic addition of new functionality, and additional work on scheduling of conflicting tasks.

Acknowledgement

This work is supported in part by Japan Society for Promotion of Science (JSPS), with the basic research program (C) (No. 20510141), Grant-in-Aid for Scientific Research.

References

1. Hulaas, G., Binder, A.V.W.J.: Portable resource control in the j-seal2 mobile agent system. In: Proceedings of International Conference on Autonomous Agents, pp. 222–223 (2001)
2. Kambayashi, Y., Takimoto, M.: Higher-order mobile agents for controlling intelligent robots. *International Journal of Intelligent Information Technologies* 1(2), 28–42 (2005)

3. Takimoto, M., Mizuno, M., Kurio, M., Kambayashi, Y.: Saving energy consumption of multi-robots using higher-order mobile agents. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 549–558. Springer, Heidelberg (2007)
4. Nagata, T., Takimoto, M., Kambayashi, Y.: Suppressing the total costs of executing tasks using mobile agents. In: Proceedings of the 42nd Hawaii International Conference on System Sciences, IEEE Computer Society CD-ROM, Los Alamitos (2009)
5. Pomerleau, D.A.: Defense and civilian applications of the alvinn robot driving system. In: Proceedings of 1994 Government Microcircuit Applications Conference, pp. 358–362 (1994)
6. Pomerleau, D.A.: Alvin: An autonomous land vehicle in a neural network. In: Advances in Neural Information Processing System, vol. 1, pp. 305–313. Morgan Kaufmann, San Francisco (1989)
7. Murphy, R.R.: Introduction to AI robotics. MIT Press, Cambridge (2000)
8. Parker, L.E.: Alliance: An architecture for fault tolerant multirobot cooperation. IEEE Transaction on Robotics and Automation 14(2), 220–240 (1998)
9. Satoh, I.: A mobile agent-based framework for active networks. In: Proceedings of IEEE System, Man and Cybernetics Conference, pp. 71–76 (1999)

Cooperation of Hybrid Agents in Models of Manufacturing Systems

František Čapkovič*

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovak Republic
Frantisek.Capkovic@savba.sk

Abstract. The cooperation of hybrid agents is synthesized in this paper and applied to continuous manufacturing system. The hybrid agent makes a cooperation of continuous and discrete event devices possible. First Order Hybrid Petri Nets (FOHPN) are utilized in order to model the elementary autonomous hybrid agents. The cooperation of hybrid agents is based on the DES (discrete-event systems) control theory. It is realized by means of DES supervision methods and the agent negotiation. The proposed approach is illustrated in details on the cooperation of hybrid production lines in a real continuous manufacturing systems.

Keywords: Agents, control, cooperation, discrete-event systems, continuous systems, hybrid Petri nets, supervisor, synthesis.

1 Introduction and Preliminaries

Continuous systems (CS) are usually described by means of differential equations describing processes at respecting physical laws. However, there are certain complex CS where it is practically impossible or very difficult to obtain a CS model corresponding to the real system. Two main difficulties occur on that way: (i) how to determine the kind of differential equations describing the particular CS - namely, to guess an order of the system when the linear differential equations are used or a kind of nonlinear differential equations; (ii) how to identify all parameters of the chosen kind of differential equations describing the real complex system by means of measuring (if any such parameters are measurable). On that account other methods are found. Hybrid Petri nets (HPN) [8] are frequently used for modelling complex hybrid systems (HS). Especially, the First Order Hybrid Petri Nets (FOHPN) [1, 9, 14] seem to be very suitable for HS modelling. HPN in general combine continuous Petri nets (CPS) [8, 9] with different kinds of Petri nets (PN) - like place/transition Petri nets (P/T PN), deterministic timed PN, stochastic PN, etc. Because any CS has minimally two discrete states - it is either working or idle - the mutual transitions between these states are, as matter of fact, the discrete events. Thus, the idea arose: (i) to create the model of

* Partially supported by the Slovak Grant Agency for Science VEGA under grant # 2/0075/09 and by the Slovak Grant Agency APVV under grant VMSP-P-0169-09.

hybrid autonomous agents (e.g. production lines in complex manufacturing systems) by means of FOHPN; (ii) then to synthesize the cooperation of the agents by means of discrete-event systems (DES) supervision. Namely, DES are driven by discrete events and the cooperation of hybrid agents exhibits attributes of DES. The idea is proposed and tested here, in this paper.

1.1 First Order Hybrid Petri Nets (FOHPN)

HPN in general are an extension of standard PN. They are able to model the coexistence of discrete and continuous variables. Such a procedure brings following advantages: (i) reducing the dimensionality of the state space; (ii) increasing the computational efficiency of the simulation process; (iii) defining optimization problems of polynomial complexity. As to HPN structure the places, transitions and oriented arcs consist of two groups - discrete and continuous. Moreover, beside the arcs among discrete places and discrete transitions and the arcs among continuous places and continuous transitions there exist the arcs among discrete places and continuous transitions as well as among the continuous places and discrete transitions. As to HPN dynamics the discrete places and transitions handle discrete tokens while the continuous places and transitions handle continuous variables (flow rates e.g. like different kinds of material flows). The mutual interaction between these groups is possible according to prescribed rules. It can be said that FOHPN were introduced by [1]. The set P of places consists of two subsets $P = P_d \cup P_c$ where P_d is a set of discrete places (graphically represented by simple circles) and P_c is a set of continuous places (represented usually by double concentric circles). Cardinalities of the sets are, respectively, n , n_d and n_c . Analogically, the set of transitions T consists of two subsets $T = T_d \cup T_c$, where T_d is a set of discrete transitions (graphically represented by simple rectangles) and T_c is a set of continuous transitions (represented usually by double rectangles - a smaller rectangle inside of the bigger one). Their cardinalities are, respectively, q , q_d and q_c . Moreover, T_d can contain a subset of immediate transitions (like in ordinary PN) and/or a subset of timed transitions. The timed transitions express the behaviour of discrete events in time and they may be deterministic and/or stochastic. Due to ensuring qualitative properties of FOHPN (so called *well-formed nets*) the following rule has to be met: firing of continuous transitions must not influence marking of discrete places. FOHPN marking is a function assigning a non-negative integer number of tokens to each of the discrete places and an amount of fluid to each of the continuous places. To each of the continuous transition t_j an instantaneous firing speed (IFS) is assigned. IFS determines an amount of fluid per a time unit (i.e. a sort of the flow rate) which fires the continuous transition in a time instance τ . For all of the time instances τ holds $V_j^{min} \leq v_j(\tau) \leq V_j^{max}$, where *min* and *max* denote the minimal and maximal values of the speed $v_j(\tau)$. Consequently, IFS of any continuous transition is piecewise constant. An empty continuous place p_i is filled through its enabled input transition. In such a way the fluid can flow to the output transition of this place. The continuous transition t_j is enabled in the time τ iff its input discrete places $p_k \in P_d$ have marking $m_k(\tau)$ at least equal to $Pre(p_k, t_j)$

(the incidence matrices **Pre** and **Post** are well known in the PN theory - see e.g. [13]) and all of its input continuous places $p_i \in P_c$ satisfies the condition that either $m_i(\tau) \geq 0$ or the place p_i is filled. If all of the input continuous places of the transition t_j have non-zero marking then t_j is strongly enabled, otherwise t_j is weakly enabled. The continuous transition t_j is disabled if some of its input places is not filled. In general, the marking development of the continuous place $p_i \in P_c$ in time can be described by the differential equation

$$\frac{dm_i}{d\tau} = \sum_{t_j \in T_c} C(p_i, t_j) \cdot v_j(\tau) \quad (1)$$

where $v_j(\tau)$ are entries of the IFS vector $\mathbf{v}(\tau) = (v_j(\tau), \dots, v_{n_c}(\tau))^T$ in the time τ and \mathbf{C} is the incidence matrix of the continuous part of FOHPN (i.e. the matrix $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$). The differential equation holds provided that no discrete transition is fired in the time τ and all of the speeds $v_j(\tau)$ are continuous in the time τ . The IFS $v_j(\tau)$, $j = 1, \dots, n_c$, defines enabling the continuous transition t_j . If t_j is strongly enabled then it can be fired with an arbitrary firing speed $v_j(\tau) \in [V_j^{min}, V_j^{max}]$. If t_j is weakly enabled then it can be fired with an arbitrary firing speed $v_j(\tau) \in [V_j^{min}, V_j]$, where $V_j \leq V_j^{max}$. Namely, t_j cannot take more fluid from any empty input continuous place than the amount entering the place from other transitions. It corresponds to the principle of conservation of matter.

The P/T PN are used here in the process of synthesizing the hybrid agent cooperation. On that way the earlier results concerning the agents cooperation based on supervision presented in [2-6] can be utilized as well as the theory of supervision [10-12]. However, the agent negotiation based on P/T PN [7] seems to be more sophisticated.

2 FOHPN Based Model of the Production Line

Consider the recycling line producing the plastic double foil from the granulate prepared from the waste plastic. The plastic foil is used for producing plastic bags. The rough FOHPN model is displayed in Fig. 1. To distinguish continuous and discrete places as well as the continuous and discrete transitions, the continuous items are denoted by capitals. The granulate is collocated in the Holder represented by the continuous place P_4 . Thence, the fluid flows through the continuous transition T_1 to the Exhausting Machine (Exhauster) represented by the place P_1 where a big bubble is blow (in order to make producing the double foil possible). Subsequently, the double foil is drawing into the prescribed width and thickness on the Drawing Line P_2 and proceeds to the Spooling Machine P_3 where the bales of a prescribed Mass are prepared. Here, after achieving the prescribe Mass, the foil is aborted by cutting, the completed bale is withdrawn and the new bale starts to be spooled on a new spool. The completed bale proceeds to another production line where the bags are produced. There, the foil is enfolded, welds corresponding to the length of bags are performed, and the

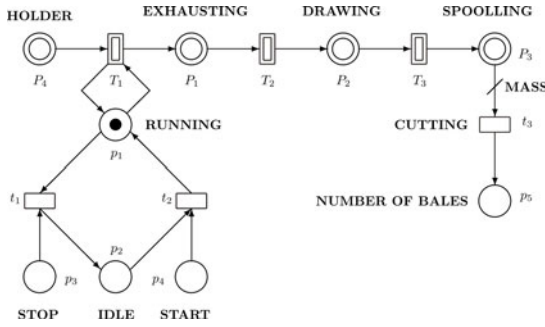


Fig. 1. The rough FOHPN-based model of the production line

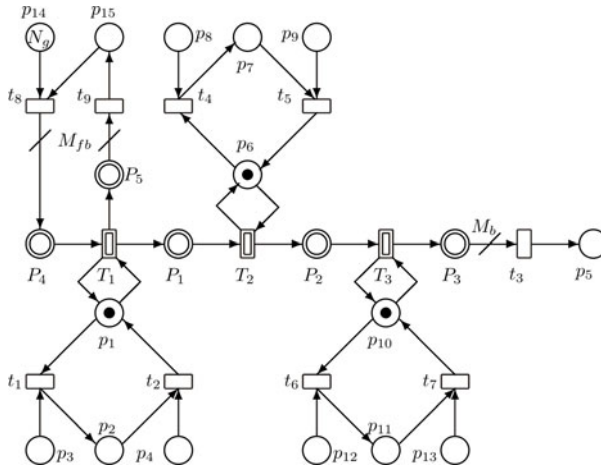


Fig. 2. The FOHPN-based model of the production line with built time delays

belt of bags is rolled into rolls with a uniform number of bags in each roll. The marking of the discrete place p_5 expresses the number of bales produced by the line. The model in Fig. 2 is more detailed than that in Fig. 1. Namely, in the real machines time delays occur (the time necessary for producing the bubble in the Exhausting Machine, the transport delay of the Drawing Line) and it is necessary to build them into the FOHPN model. The $\{p_6 - p_9, t_4, t_5\}$, $\{p_{10} - p_{13}, t_6, t_7\}$ model, respectively, the delays of exhausting and drawing processes. While $t_4 - t_7$ are timed transitions modelling delays, p_8, p_9, p_{12}, p_{13} appropriately affect their firing. Moreover, it is necessary to ensure regular supplying the granulate in order to avoid breakdown of the line caused by the lack of the granulate. This is done by means of the feedback $T_1 \rightarrow P_5 \rightarrow t_9 \rightarrow p_{15} \rightarrow t_8$. Here, M_{fb} denotes the multiplicity of the arc due to added amount of the granulate in one batch. The marking N_g of the discrete place p_{14} represents the number of the

added batches of the granulate. In addition, M_b represents here the multiplicity of the arc corresponding to the prescribed mass of the bale. The structure of the FOHPN model can be described by the following incidence matrices where the indices cc, cd, dc, dd denote, respectively, the incidences between places and transitions 'continuous-continuous', 'continuous-discrete', 'discrete-continuous', 'discrete-discrete'.

$$\begin{aligned}
 \mathbf{Pre}_{cc} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \mathbf{Post}_{cc} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\
 \mathbf{Pre}_{cd} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_b & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & M_{fb} \end{pmatrix}; \mathbf{Post}_{cd} = \begin{pmatrix} 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 & M_{fb} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 \mathbf{Pre}_{dc}^T &= \begin{pmatrix} 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 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \equiv \mathbf{Post}_{dc}^T \\
 \mathbf{Pre}_{dd} &= \begin{pmatrix} 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 \\ 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 & 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 & 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 & 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 & 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 & 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 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}; \mathbf{Post}_{dd} = \begin{pmatrix} 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 & 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 & 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 & 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 & 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}
 \end{aligned}$$

The initial marking of the continuous places is $\mathbf{M}_c = (0, 0, 0, M_{gr}, 0)^T$ where M_{gr} is the initial amount of the granulate in P_4 . The initial marking of the discrete places is $\mathbf{m}_d = (1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, N_g, 0)^T$ where N_g is the number of the batches of the granulate to be added during the production. Firing speeds of the continuous transitions $v_j(\tau)$, $j = 1, 2, 3$ are, respectively, from the intervals $[V_j^{min}, V_j^{max}]$. The discrete transitions are considered to be deterministic without any delay or with a transport delays mentioned above. Using the Matlab simulation tool HYPENS [14] with the structural parameters $M_b = 270$, $M_{fb} = 3750$, the initial markings with $M_{gr} = 5000$, $N_g = 4$, with the

limits of intervals for the firing speeds being $V_j^{min} = 0, j = 1, 2, 3, V_1^{max} = 1.8, V_2^{max} = 1.5$ and $V_3^{max} = 1.4$ and finally, with the delays of discrete transitions being the entries of the vector $(0, 0, 0, 0.01, 125, 0.01, 300, 0, 0)^T$ we obtain the behaviour of the simulated line. It is illustrated in Fig. 3 and Fig. 4. In the upper part of Fig. 3 the course of flows is shown during the time segment when the granules are added fourtimes while the lower part displays the zoomed detail.

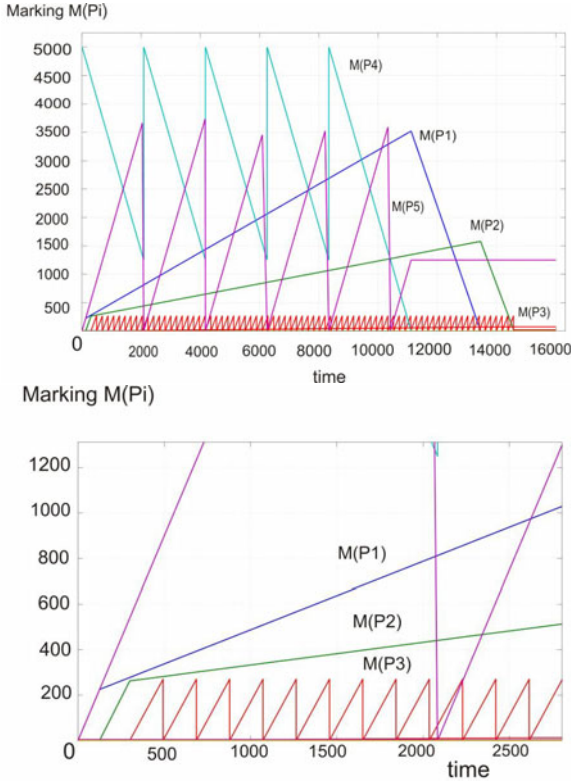


Fig. 3. The dynamics behaviour of the line material flows in the common scale (upper picture) and the zoomed detail in order to see the transport delays better (lower picture)

2.1 Description of the Simulation Results

In Fig. 3 and Fig. 4 results of the production line simulation are presented. They illustrate the dynamic development of marking $M(P_i)$ of the continuous places $P_i, i = 1 \dots, 5$, i.e. draft the courses of the material flows throughout these places. $M(P_5)$ models a feedback flow (a batch of granules is poured into the Holder in order to avoid its emptying), $M(P_4)$ the flow of granules throughout the Holder, $M(P_1)$ the flow of the exhausted bubble from the Exhauster, $M(P_2)$ the flow of the drawing foil and $M(P_3)$ spooling the particular bales of the foil. The finished

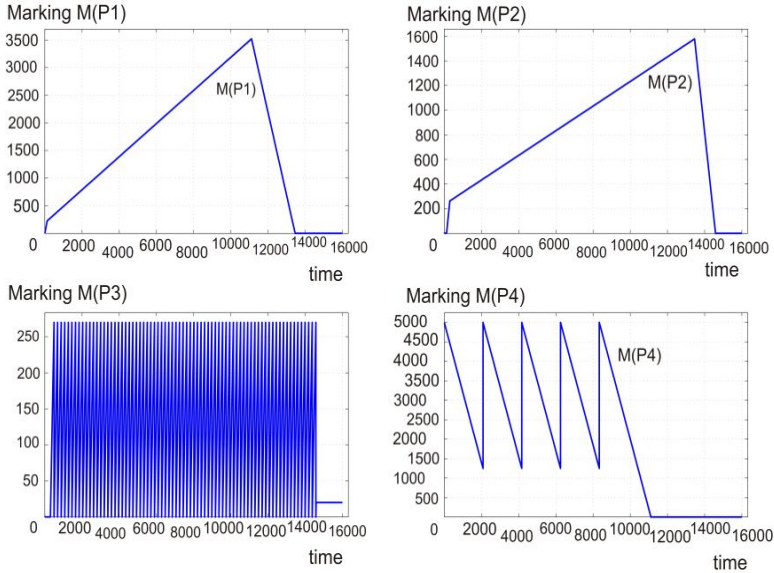


Fig. 4. The dynamics behaviour of the material flows of the production line in their individual scales - i.e. the marking evolution of the continuous places $P_1 - P_4$ in time

bale is immediately removed in order to spool the next bale. When the flow of granules is stopped, the other flows are gradually finished too.

3 Cooperation of Hybrid Agents

Having a group of hybrid agents the cooperation can be synthesized - see in [3]-[6] - by means of the DES supervision methods. Consider N lines for the foil production and $M \leq N$ lines for producing the bags rolls. These lines are hybrid and they can be understood to be hybrid autonomous agents. The lines producing the foil can work independently, even simultaneously in time. The same is valid for the lines producing the rolls of bags. However, N lines producing the foil have to share only M rolling lines. To solve this problem a strategy has to be defined for the sharing. When such a strategy is defined the conditions for the supervisor synthesis can be formulated and the corresponding supervisor can be synthesized. The simplest form of the agents cooperation is in case when there exists a buffer at any line producing the foil. The lines can produce foil and store the bales into their buffers. The idle rolling lines can take the bales of foil from the buffers as they want. A form of scheduling seems to be a more sophisticated strategy which makes an optimizing possible - e.g. minimizing the idle time. However, it depends also on actual needs of foil parameters (especially on its width and thickness). However, the negotiation based on the offers and demands

seems to be the most sophisticated strategy. The P/T PN-based model can be usable also on that way - see [7].

3.1 The Model of P/T PN in Analytical Terms

Namely, P/T PN yield the following linear discrete system (in the sense of system theory) applicable for describing the communication of agents

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k \quad , \quad k = 0, \dots, N \quad (2)$$

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} \quad (3)$$

$$\mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k \quad (4)$$

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; $\sigma_{p_i}^k \in \{0, 1, \dots, c_{p_i}\}$, $i = 1, \dots, n$ express the states of the 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 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; its components $\gamma_{t_j}^k \in \{0, 1\}$, $j = 1, \dots, m$ represent occurring of 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 matrices of integers; $\mathbf{F} = \{f_{ij}\}_{n \times m}$, $f_{ij} \in \{0, M_{f_{ij}}\}$, $i = 1, \dots, n$, $j = 1, \dots, m$ expresses the causal relations among the states (as causes) and the discrete events occurring during the DES operation (as consequences) by 0 (nonexistence of the 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 (as causes) and the DES states (as consequences); the matrix \mathbf{B} is given by means of the arcs incidence matrices \mathbf{F} and \mathbf{G} according to (3); $(\cdot)^T$ symbolizes the matrix or vector transposition.

3.2 The Negotiation of Agents

The example of negotiation of a couple of agents A_1 , A_2 can be described as follows. Consider the following interpretation of P/T PN places: p_1 - A_1 does not want to communicate; p_2 - A_1 is available; p_3 - A_1 wants to communicate; p_4 - A_2 does not want to communicate; p_5 - A_2 is available; p_6 - A_2 wants to communicate; p_7 - communication; and p_8 - availability of the communication by means of the interface (a communication channel). The interpretation of most of the PN transitions is clear, but it is necessary to point out some of them: t_9 - fires the communication when A_1 is available and A_2 wants to communicate with A_1 ; t_{10} - fires the communication when A_2 is available and A_1 wants to communicate with A_2 , and t_{12} - fires the communication when both A_1 and A_2 wants to communicate each other. The model is presented in Fig. 5. Here, as it is clear, the interface realizing the negotiation process has the form of the PN

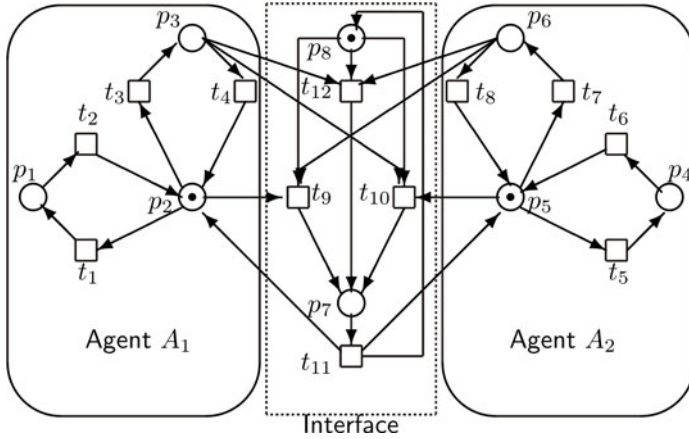


Fig. 5. The P/T PN model of the negotiation process

module (PN subnet). The P/T PN based model of the agents communication has the following parameters

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{A_1} & \mathbf{0} & \mathbf{F}_{A_1-A_2} \\ \mathbf{0} & \mathbf{F}_{A_2} & \mathbf{F}_{A_2-A_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{neg} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & | & 0 & 0 & 0 & 0 & | & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & | & 0 & 0 & 0 & 0 & | & 0 & 1 & 0 & 1 \\ - & - & - & - & | & - & - & - & - & | & - & - & - & - \\ 0 & 0 & 0 & 0 & | & 0 & 1 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & | & 1 & 0 & 1 & 0 & | & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 1 & | & 1 & 0 & 0 & 1 \\ - & - & - & - & | & - & - & - & - & | & - & - & - & - \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{G}^T = \begin{pmatrix} \mathbf{G}_{A_1}^T & \mathbf{0} & \mathbf{G}_{A_1-A_2}^T \\ \mathbf{0} & \mathbf{G}_{A_2}^T & \mathbf{G}_{A_2-A_1}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{neg}^T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ - & - & - & - & | & - & - & - & - & | & - & - & - & - \\ 0 & 0 & 0 & 0 & | & 1 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & | & 0 & 1 & 0 & 1 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 1 & 0 & | & 0 & 0 & 0 & 0 \\ - & - & - & - & | & - & - & - & - & | & - & - & - & - \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & | & 0 & 0 & 1 & 0 \end{pmatrix}$$

To apply the model with these parameters it is necessary to choose an initial state. Of course, modelling of more cooperating agents in such a way is possible too.

4 Conclusion

The hybrid agents were defined and the approach to synthesis of their cooperation in models of manufacturing systems was pointed out. The individual agents were modelled by means of FOHPN while the cooperation of the agents was modelled by P/T PNs. The instance of the concrete practical application - the recycling line producing the plastic double foil - was introduced in order to underline soundness of the approach. By means of simulating the production line dynamic behaviour in Matlab the satisfying applicable results were obtained.

References

1. Balduzzi, F., Giua, A., Menga, G.: Modelling and Control with First order Hybrid Petri Nets. *IEEE Trans. on Robotics and Automation*. 16, 382–399 (2000)
2. Čapkovič, F.: Modelling, Analysing and Control of Interactions among Agents in MAS. *Computing and Informatics* 26, 507–541 (2007)
3. Čapkovič, F.: Automatic Control Synthesis for Agents and their Cooperation in MAS. *Computing and Informatics* 29, 1045–1071 (2010)
4. Čapkovič, F.: Cooperation of Agents in Manufacturing Systems. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6070, pp. 193–202. Springer, Heidelberg (2010)
5. Čapkovič, F.: DES Control Synthesis and Cooperation of Agents. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009*. LNCS, vol. 5796, pp. 596–607. Springer, Heidelberg (2009)
6. Čapkovič, F.: A Modular System Approach to DES Synthesis and Control. In: *2009 IEEE Conference on Emerging Technologies & Factory Automation - ETFA 2009*, 8 pages, CD ROM. IEEE Press, Piscataway (2009)
7. Čapkovič, F., Jotsov, V.: A System Approach to Agent Negotiation and Learning. In: Sgurev, V., Hadjiski, M., Kacprzyk, J. (eds.) *Intelligent Systems: From Theory to Practice*. SCI, vol. 299, pp. 133–160. Springer, Heidelberg (2010)
8. David, R., Alla, H.: On Hybrid Petri Nets. *Discrete Event Dynamic Systems: Theory and Applications* 11(1-2), 9–40 (2001)
9. Dotoli, M., Fanti, M.P., Giua, A., Seatzu, C.: First-order hybrid Petri nets. An application to distributed manufacturing systems. *Nonlinear Analysis: Hybrid Systems* 2(2), 408–430 (2008)
10. Iordache, M.V., Antsaklis, P.J.: Supervision Based on Place Invariants: A Survey. *Discrete Event Dynamic Systems* 16, 451–492 (2006)
11. Iordache, M.V., Antsaklis, P.J.: *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. Birkhauser, Boston (2006)
12. Iordache, M.V.: *Methods for the supervisory control of concurrent systems based on Petri nets abstraction*. Ph.D. dissertation, University of Notre Dame, USA (2003)
13. Murata, T.: Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE* 77(4), 541–580 (1989)
14. Sesseggo, F., Giua, A., Seatzu, C.: HYPENS: A Matlab Tool for Timed Discrete, Continuous and Hybrid Petri Nets. In: van Hee, K.M., Valk, R. (eds.) *PETRI NETS 2008*. LNCS, vol. 5062, pp. 419–428. Springer, Heidelberg (2008)

Multi-Agent Oriented Integration in Distributed Control System

Dariusz Choinski and Michał Senik

Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{dariusz.choinski,michal.senik}@polsl.pl

Abstract. This paper introduces a possibility of integration of Distributed Control System (DCS) by Multi-Agent System (MAS). Proposed solution is based on Java Agent Development (JADE) framework a Java based tool that supports creation of FIPA (Foundation for Intelligent Physical Agents) compliant agents. In the proposed Multi-Agent heterogeneous system the OPC plays an important role as a powerful communication technology especially dedicated for real-time DCS. Presented agents are capable of sharing their knowledge with other agents; they are capable of accomplishing tasks according to actual environment conditions. Proposed solution of the MAS implementation is a highly efficient, scalable and robust it utilizes mobility in order to ensure cooperativeness during distributed problem solving task. Finally, it has been proved that assumed Multi-Agent System considerably improves system quality, accuracy and reliability.

Keywords: Multi-Agent Systems, hybrid systems, Distributed Control Systems, FIPA, concurrent programming, JADE, OPC.

1 Introduction

DCS is a very wide term that can be applied to various types of industries which are very complex, decentralized and dynamic with large amounts of various data that needs to be analyzed concurrently. Fast and accurate integration of such a complex and decentralized system is therefore not a trivial task because it requires extremely wide area of expertise. Nonetheless, it is not impossible because DCSs have been present in automation industry for a certain period of time and have already become a standard both as a concept and methodology.

MASs however are a quite new concept but already they proved to be very promising especially because of their ability of self independence, interoperability and possibility of fast and concurrent way of solving real time problems. In fact, MAS and DCS share similar if not the same philosophy. Both are dynamic, decentralized and redundant. MASs are designed specifically to support DCSs in many areas. Concurrent integration of such systems fits the profile of MASs. MAS requires a way in which it can cooperate with its DCS. OPC automation protocol [1] is one such possibility. It is a very flexible protocol which improves fast integration. OPC has become a standard that is used by many third party vendors to integrate both their hardware

and software. This paper deals with further solutions of hierarchical and multi-agent control and information system with real-time update and self organisation presented for example in [2]. The proposed hierarchical structure was also mentioned and described in [3].

The DCS architecture and software is based on rules for structuring and formalism of the technological projects [5]. The structure defines dependencies and relations between objects that are considered in the process of designing, construction, realising, operation, exploitation and disposal. The particular structures are organized hierarchically and should specify the information regarding the system, content of the formalized documents and composition of referential designations.

2 Multi-Agent System

There are many solutions of how MAS can be structured [6]. Prior to MAS implementation automation system granulation should be established. Granulation of the system shows how agents should correspond to system resources. Such knowledge verifies weight of the source code that corresponds to each agent. This also tells about particular agent functionalities.

To some extent, MAS relieves the user from both hard analysis and decision making processes making such systems less error prone. Such system notion is focused around ontology concept. In most cases ontologies are considered as system knowledge. This knowledge is modeled and implemented beforehand MAS application and it is known prior to its runtime [7], [8]. Ontology describes static properties of the system as well as runtime states in which the system can reside and conditions of transitions between those states. Based on the ontology, agents can share their common knowledge and work together over common system problems because sharing knowledge is the natural way for single pieces of MAS software to achieve their goals. Deeper analysis reveals that ontology is a set of simple and more complex rules such as concepts, predicates and actions entities. Those entities can be organized in a hierarchical structure which means that simpler entities can be nested in more complex entities. Such approach allows for dynamic management of system knowledge using its static descriptions.

2.1 Knowledge Sharing Based on Hybrid System

Knowledge sharing based on ontologies look simple for discrete systems because of being based on discrete logic [9]. However, for continuous processes with infinite number of states, when the problem is more difficult, creating ontology may still be possible in some cases [10], [11]. Using description with decomposition to subsystems and functions, it is possible to treat subsystems as hybrid automata [12], [13] and to apply flexible control of hybrid systems [2]. The major advantage of a hybrid system deals with its particular proprieties when event-driven part (hybrid automaton subsystem) and time-driven part (continuous state equations subsystem) can be corrected separately without disturbing the whole system. Hybrid control facilitates recognition of discrete states required by technology. If it is possible to define those states and technology rules, than binding of those states by ontology is also possible.

Practice of processes automation shows that the automated plant may be represented as hybrid state machine [2], [14]. A-prior knowledge about the process is based on rules of the hybrid systems with explicit models. One of the important features describing such systems refers to the description of condition for transitions between states.

Proposed Hybrid System consists of two subsystems: one representing a continuous dynamic and the other representing discrete events and mutual interactions with other systems. Those conditions are given either explicitly – are computed beforehand based on the knowledge about the plant and are dependent on a measurable state, or implicitly - are given by differential and algebraic equations describing a divisible state space and transitions specifications [2]. Those equations must be solved on-line [15], [4]. In both cases, knowledge about a continuous controlled plant (as a whole) may be approximated in order to use discrete control on the higher level of the hierarchy. Control of a hybrid automaton is a complex task, which can be achieved with MAS. The proposed structure of control agents is hierarchical and consists of three layers: control, supervisory and expert. Each layer performs tasks at a designated level of abstraction, and actively cooperates to achieve the control goal in the most efficient manner [2].

2.2 Multi-Agent System Application

AFIOPC (Agents for Integration of OPC) platform is a real time processing, hierarchical, MAS capable of dynamical adjustment to existing information structure [16]. It wraps over existing JADE framework using a set of newly implemented agents thus creating a complex system. Its existence is dictated by today's constantly arising problem of having rapidly changing environment, which should be handled with extreme caution in order to maintain high level of quality of acquired information. AFIOPC platform is designed especially to be responsible of dealing with a situation of having probability of such dynamic changes. It can function on one or more remotely controlled hosts, which may vary in time. This indicates that AFIOPC platform can adapt to almost any existing situation

The AFIOPC platform establishes efficient cooperation with OPC Servers through Java Native Interface (JNI) which bridges Java mechanisms into the native language world such as C/C++ or Delphi. OPC is an open automation interface protocol for fast data retrieval and analysis and it is being actively developed by OPC Foundation since 1994. Until now OPC Foundation introduced a set of different communication standards amongst which the most widely used is OPC Data Access (OPC DA) interface which is used for retrieval of real time data [1].

3 JADE OPC Agent Platform

In the current development stage AFIOPC platform consists of eight different types of agents: Management Agent (MA), Network Scanner Agent (NSA), Port Scanner Agent (PSA), Discovery Agent (DA), OPC Agent (OA), Lifecycle Supervisory Agent (LSA), Supervisory Agent (SA) and Node Agent (NA). Those different types of agents can interact with each other in several ways. Each AFIOPC platform agent

shares similar capabilities making them more compliant and consistent. Those capabilities are the key concepts of the internal platform architecture and involve mobility, cloning, common messages, behaviors, threading, GUI, ontology, event subscription, message notifications and message propagation mechanisms.

The AFIOPC platform ontology bridges OPC interface protocol specification into AFIOPC MAS internal mechanisms. Moreover, it links this information into higher level ontological structures which can be analyzed by each single AFIOPC agent. It has been created to not only describe static system characteristics but also to be used whenever OPC interface protocol native asynchronous readouts are performed i.e. server status, tags properties, tags values triplets (value, quality and timestamp) concurrently. By means of AFIOPC platform ontology each platform agent is able to share its current knowledge state with other listeners. This stands as a natural agent-based knowledge decomposition problem. Moreover, ontology is extensively used during system knowledge validation.

3.1 Agents Concepts

To set up the AFIOPC platform a Main Container (MC) must be created. This MC hosts initially two major agent entities DF (Directory Facilitator) and AMS (Agent Management System). MC with AMS and DF agents are the only architectural prerequisites. Each platform agent at the beginning of his life registers itself with the DF so it can be discovered by the other agents that wish to utilize its capabilities. Moreover each platform agent besides the fact that can be initiated manually from the JADE GUI can also be started remotely, by the other, already alive agents. Scenarios of how platform agents will be born or die are in this case undeterminable. It is worth to mention that agents run always in separate thread and each single agent behavior is executed in separate thread also. Such approach enables asynchronous and concurrent agent cooperation with many other different types of platform agents. Each described agent operation is performed via customized set of concurrent behaviors. Each behavior extends common mechanism of knowledge sharing that utilizes AFIOPC platform ontology. This is a common agent engine that supports concurrent analysis of multiple ontological expressions.

Complete control over AFIOPC platform is established by SA. This agent is a solution specific entity by means of which other platform specific agents can be managed. By default it is located inside the MC but it is not required. This is an implementation issue only and can be changed in the future. There can be more than just one SA present in a single agent platform and it is not required that each one should be located inside a different platform container. There is no conflict between multiple SA if they are present in the same container at the same time.

PSA is responsible for port scanning processes. It discovers enabled ports on the remote machine and transfers this information directly to the SA for further processing.

NSA is responsible for acquiring network specific pieces of information. In short time it can discover remote host locations that are available in the network in which the AFIOPC platform operates. It supplies SA with that particular data. It is brought to life by SA during its initialization. When active, it asynchronously checks a desired pool of addresses that the listener wants to check. NSA does not wait until all

addresses in the pool are checked and it immediately returns all valid host specific data to the listener. NSA, after IP pool scan completion does not terminate but simply starts the scan process again after a short break.

DA is responsible for remote host environment discoveries. It is brought to life by SA just after NSA sends back first notification messages or right after NA wakes up. It acquires data about possible presence of OPC Server services. It can perform its task remotely or it can travel to the desired location and send back obtained results. DA is also responsible for OA creation. It discovers available OPC Server services and creates the exact number of OA entities. Each created OA is then connected with a single OPC Server service from the discovered OPC Server pool. DA relieves user from performing OPC Server search, connection, status, structure and tags properties retrieval operations. It is also responsible for maintaining the exact OA number on single host connected to the AFIOPC platform. If for some unexpected reason one OA terminates, DA tries to reestablish such connection again.

OA is responsible for acquiring data from OPC Server services (Fig. 1). It can configure OPC Server with specific OPC Groups and OPC Items. It can easily reconnect between various types of OPC Servers but it interacts with only one. It can work with a local and remote OPC Server services. It can also move between different remote locations to reach the desired OPC Server and interact with it locally. It can send back all the obtained data back to various types of listeners. Such notification process can occur between single or dynamic number of other platform agents. OA during its start registers itself in the agent platform so it can be recognized by other agents that wish to utilize services this OA provides. OA can speak with a dynamic number of other platform agents concurrently. That way knowledge propagation inside the AFIOPC platform occurs really fast. Generally speaking, information propagation is a strong side of the AFIOPC platform; each agent shares the same generic algorithm of data propagation.

LSA is closely coupled with OA because of the possible instability of the OA connection with third party OPC Services and a native code itself. OA JVM process instability during OPC Server interaction is a very rare phenomenon but it has been observed that when it happens it has very destructive consequences to the platform itself. It can be even demonstrated that lighter JVM instability can crash single JVM process. Therefore LSA existence is justified. It checks whether particular platform OA is available or not and if not restores it back to its nominal work state based on the latest terminated agent configuration. There is a redundancy between OA, LSA and DA. Unexpected OA process crash will be detected either by its LSA or DA.

MA is responsible for user interaction with AFIOPC platform resources. It is an agent with GUI which allows user real time data manipulation. There can be many different MAs running inside AFIOPC platform. Each one will receive notification from the different platform agents based on services they provide. MAs are designed to configure various OAs concurrently. Each OA when configured notifies registered MAs with its configuration and fresh readouts.

NA is responsible for registering remote hosts to the AFIOPC platform. It has a straightforward concept of operation and when active it joins its host with the running AFIOPC platform.

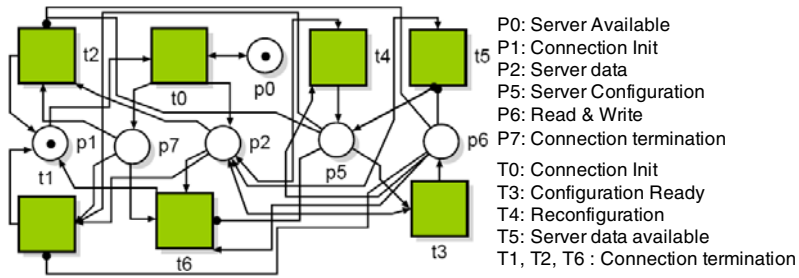


Fig. 1. OPC Agent OPC Server interaction - graph with initial marking

3.2 Agents Asynchronous Concurrent Cooperation

Idea of the AFIOPC platform as DCS is focused around concurrent cooperation between various factory floor OPC Server services and high level human decision making and data analysis processes. Because of that, each platform agent has been equipped with asynchronous mechanisms that allow it to communicate simultaneously with other agents inside the same or different agent platform. That way, communication process is fast and reliable not only between agents themselves but between a human user also.

AFIOPC platform can function not only as a stand alone agent platform but can also communicate with numerous other agent platforms concurrently. Agents from one platform can utilize knowledge that is already available for agents located inside different agent platform.

3.3 Experimental Agent Platform Integration Scenario

Initially, only one workstation with MC, AMS and DF agents is available. Other workstations, with number of different OPC Server entities will pop up in the system dynamically. Proposed integration problem assumes single AFIOPC platform sufficient (Fig 2.).

After platform initialization (P0, T0) all required JADE components are initialized (MC with DF and AMS agents). When this stage is complete whole platform is considered to be in the running state (P1). Platform can be terminated at any point of time, whenever it is requested (T1, T10, T11 or T12). Transition T1 is enabled just after platform initializes, T10 is enabled just after a set of initial agents are initialized (SA, MA or NA agents), T11 transition is enabled when first DAs pops up in the platform, T12 transitions enables just after first OAs registers to the platform.

Transitions T1, T10, T11 and T12 exclude each other. Platform termination process varies in the different states. Transition T9 initializes AFIOPC platform main components such as SA (P5). At this point NA (P6) can arrive at any time as well as MA (P11). Whenever new NA initialize it register with platform DF agent so it can be discovered by other agents i.e. SA. Whenever that happen new remote host joins the platform providing its resources to be utilized (T5). When transition T5 fires, new DA shows up. DA searches for all available OPC Server services (P8) and assigns (T7) separate OA to each. It tries to hold this dynamic pool of OAs according to OPC

Server services pool (P8). If for some unexpected reason one of the discovered OPC Servers is discarded from the pool its OA is terminated along with its LSA (T8). DA avoids any further attempts to recreate this OA unless OPC Server will be available again. AFIOPC platform terminates (P2) when no platform specific agent is available (T2).

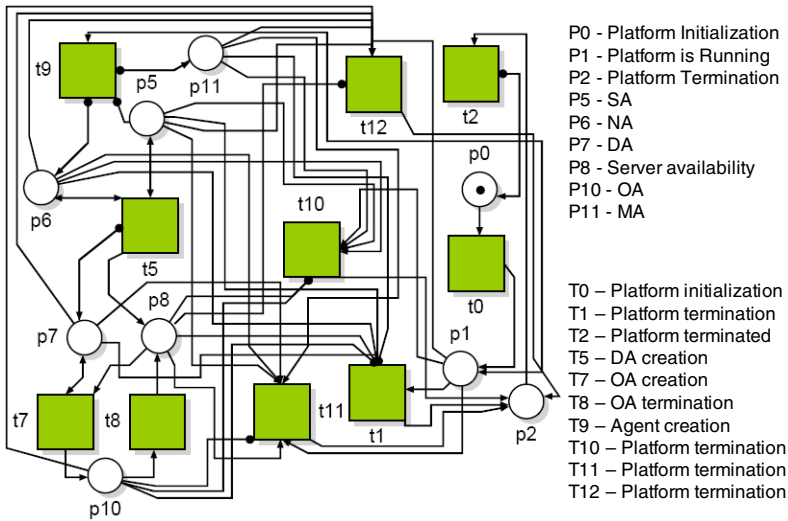


Fig. 2. JADE OPC Agent platform lifecycle – Petri graph with initial marking

3.4 Integration Scenario Simulation Results

Based on presented experimental agent integration scenario real time AFIOPC simulations have been conducted. Simulations were performed on two remote workstations which host several different OPC Server services as well as number of MAs which were used to perform concurrent configuration. During simulation four different scenarios has been reworked (Table 1.).

Each remote location is registered in the system by NA. Having deployed DA agents onto available locations multiple OAs and their LSAs started to manifest their presence in the platform with OPC Server configuration data and agent specific data. Number of OPC Server connections which also corresponds to number of OAs and

Table 1. AFIOPC simulation configurations

AFIOPC simulation configurations				
Sim. no.	Host no.	Server no.	Tag no.	Time interval / Refresh rate [ms]
1	1	5	461	5000
2	2	12	1033	5000
3	2	20	1934	5000
4	2	20	1934	1000

Table 2. AFIOPC simulation configuration interacting agents

AFIOPC simulation configurations - agents							
Sim. no.	MA no.	OA no.	LSA no	DA no.	SA no.	NA no.	Overall
1	1	5	5	1	1	1	14
2	2	12	12	2	1	1	28
3	3	20	20	2	1	1	47
4	3	20	20	2	1	1	47

LSAs, has been deliberately changed during simulations to gain access to various different OPC tags. Configuration data upload automatically invokes OPC Server readouts. Fresh data arrives at timely configured intervals.

OPC Server specific data (Table 2.) are always hierarchically structured and can be divided into status data, namespace structure data, tags properties data and readouts. Agent specific data are flat and applies to container name, address, port and ID, host IP, agent name and address. Agent specific data are mostly static though can unexpectedly change because of the possible relocation or disposal of the agent or whole container. OPC Server specific data on the contrary is mostly dynamic apart from structure or properties data.

Table 3. OPC Server specific data

	OPC Server specific data	
	Dynamic	Static
status	connection state, current time, last update time	Vendor, start time
structure	---	Hierarchical, multilevel OPC leafs & branches structure
tags properties	timestamp	OPC tag fully qualified name (FQN), canonical data type, access rights, engineering units (EU), scan rate, description etc.
readouts	Value, quality, timestamp	---

Amount of data that agents exchange between OPC Server services (Table 3.) and each other brought additional network workload problems which can seriously affect overall quality of the obtained results. This problem however has been solved because collaborating agents memorizes other agent’s states. Such internal mechanism minimizes consequences of the possible network overload because it allows sending only needed data, which are considerably smaller, during multiple update notifications. It has been observed that reducing configured time interval may seriously affect AFIOPC during run time. This is true whenever amount of data is huge (Table 1. simulation no. 4). To simulate OA instability and future failure system task manager process removal task has been performed. OA process sudden death is almost alike to OA JVM process internal failure. This situation is handled by means of two agents and is indeterministic because it can be performed either by DA or OA’s LSA. Each time OA process has been abruptly terminated one of mentioned agent restored it with the last known configuration.

4 Conclusions

The presented MAS proves to be a highly productive solution considering DCS conditions and requirements because both systems share the same philosophy. Both can dynamically adapt to changing environment, both are redundant and concurrent by their nature. Solution of the presented MAS is based on Java technology. Efficiency of the presented system is strengthened mainly because of the generic, solution specific agent mechanisms that have been developed during the beginning phase of the implementation process. Key benefits of this system lay in durability, scalability, robustness, concurrency, redundancy, safety, self-awareness level and self-independence level.

Performed simulation scenarios proves that MAS approach to DCS range of problems in which multiple OPC Server services need to be handled is very efficient. Simulation results reveal great potential of the presented solution. Presented solution is a FIPA compliant platform that means that its structure is open to the newly created agents. This means that new agents with fresh functionality can cooperate and utilize functionality of agents that already exists. It is a great advantage considering the fact that both environment and requirements can dynamically change. It also means that exiting agent functionality can be reused by the newly created agents rather than rewritten which would be a waste of time. In addition to that it is worth to mention that in the situation in which the only right solution would be agent reimplementaiton it would still be more efficient. It is because we are not changing the whole existing system, we are altering one piece of it which minimizes the possibility of errors in the rest of the environment. Agency minimizes the possibility of introducing new errors to the parts of the system which should stay unchanged.

MASs for some time have been introduced as an area of extensive research. This approach proves to have a lot of hidden power and abilities that previously have been omitted possibly by mistake, misjudgment or misinformation. The AFIOPC platform is one such example of MAS which deals with a varying number of OPC Server services via OPC DA interface protocol. Functionality of the AFIOPC platform has been introduced along with its current development stage capabilities. Exhaustive tests have been also introduced as they show robustness; efficiency and easiness of MAS based approach to knowledge management. The AFIOPC platform proves that it is a reliable system that is capable of fast and concurrent retrieving and analysis of large amounts and various kinds of information asynchronously.

Processes of agent data sharing extensively utilize aspect of common analysis of ontological structures. Those structures play a major role in agent communication act as well as determine each agent behavior.

Further area of research will be directed towards autonomous and cooperative aspects of MASs. Those aspects will include research in the areas of errors in analysis and backtracking, independent decision making processes as well as common cooperation with end user. The current and future effort will be targeted at the negotiating protocols based on the rules of hybrid systems with explicit models and its implementation.

Acknowledgments. This work was supported by the Polish Ministry of Scientific Research and Higher Education grant no. N N514 471539 and by the Silesian University of Technology BW/RGH-2/Rau0/2010.

References

1. Iwanitz, F., Lange, J.: *OPC – Fundamentals, Implementation and Application*. Huthig Verlag, Heidelberg (2006)
2. Choński, D., Nocoń, W., Metzger, M.: Multi-Agent System for Hierarchical Control with Self-organising Database. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2007. LNCS (LNAI)*, vol. 4496, pp. 655–664. Springer, Heidelberg (2007)
3. Tan, V.V., Yoo, D.-S., Shin, J.-C., Yi, M.-J.: A Multiagent System for Hierarchical Control and Monitoring. *Journal of Universal Computer Science* 15, 2485–2505 (2009)
4. Czczot, J., Laszczyk, P., Metzger, M.: Local balance-based adaptive control in the heat distribution system - Practical validation. *Applied Thermal Engineering* 30, 879–891 (2010)
5. Wainwrighta, D., Waring, T.: Three domains for implementing integrated information systems: redressing the balance between technology, strategic and organisational analysis. *International Journal of Information Management* 24, 329–334 (2004)
6. Chira, O., Chira, C., Roche, T., Tormey, D., Brennan, A.: An agent-based approach to knowledge management in distributed design. *J. Intell. Manuf.* 17, 737–750 (2006)
7. Choński, D., Metzger, M., Nocoń, W.: Multiscale Three-Phase Flow Simulation Dedicated to Model Based Control. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloat, P.M.A. (eds.) *ICCS 2008, Part II. LNCS*, vol. 5102, pp. 261–270. Springer, Heidelberg (2008)
8. Metzger, M.: Comparison of the RK4M4 RK4LIN and RK4M1 methods for systems with time-delays. *Simulation* 52, 189–193 (1989)
9. Buse, D.P., Wu, Q.H.: *IP Network-based Multi-agent Systems for Industrial Automation*. Springer, Heidelberg (2007)
10. Kularbphetpong, K., Clayton, G., Meesad, P.: A Hybrid System based on Multi-Agent System in the Data Preprocessing Stage. *International Journal of Computer Science and Information Security* 7, 199–202 (2010)
11. Nocoń, W., Metzger, M.: Predictive Control of Decantation in Batch Sedimentation Process. *AICHE Journal* 56, 3279–3283 (2010)
12. Sure, Y., Gomez-Perez, A., Daelemans, W., Reinberger, M.-L., Guarino, N., Noy, N.F.: Why Evaluate Ontology Technologies? Because It Works! *IEEE Intelligent Systems* 19, 74–81 (2004)
13. Sauer, O., Sutschet, G.: A step towards real time [production monitoring and control system]. *Manuf. Eng.* 85, 32–37 (2006)
14. Akkermans, H., Gordijn, J.: Ontology Engineering, Scientific Method and the Research Agenda. In: Staab, S., Svátek, V. (eds.) *EKAW 2006. LNCS (LNAI)*, vol. 4248, pp. 112–125. Springer, Heidelberg (2006)
15. Solskinnsbakk, G., Gulla, J.A.: Ontological Profiles in Enterprise Search. In: Gangemi, A., Euzenat, J. (eds.) *EKAW 2008. LNCS (LNAI)*, vol. 5268, pp. 302–317. Springer, Heidelberg (2008)
16. Leduc, R.J., Lawford, M., Dai, P.: Hierarchical Interface-Based Supervisory Control of a Flexible Manufacturing System. *IEEE Transactions on Control Systems Technology* 14, 654–668 (2006)

pPDC Blackboard Broadcasting in Agent-Based Distributed Process Control

Grzegorz Polaków and Mieczyslaw Metzger

Faculty of Automatic Control, Electronics and Computer Science
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
{grzegorz.polakow,mieczyslaw.metzger}@polsl.pl

Abstract. There is an increasing demand in modern industrial process control for advanced control strategies e.g. sustaining controlled oscillations, which cannot be performed with classical determined control algorithms. At the same time, the class of continuous industrial processes is well-fitted to agent-based control, where the state of the control system is held and processed as a blackboard. A framework is proposed to support such agent-based advanced control systems. The framework is based on the idea of cyclical blackboard broadcasting in order to keep the distributed control system synchronised. The primary component of the framework is the specifically developed networking protocol, based on a real-time architecture adopted from the industrial informatics domain. The approach ensures strict time-determinism of the inter-agent communication, which is required for continuous process control but is not supported in the already existing multi-agent environments.

Keywords: continuous processes, hybrid control, agent-based control, agent framework, agent-based biooscillations stimulation.

1 Introduction

In [1] an original pPDC (parallel Producer – Distributor – Consumer) communication architecture intended for applications in industrial manufacturing and robotics was presented. The pPDC protocol is designed to share a single main resource, which is broadcasted over a network in order to synchronise connected peers. The very same protocol may also be employed as a basis for an agent environment for a specific class of multi-agent systems i.e. intelligent control systems for industrial processes.

The classic approach to control in industrial processes consists of employing one of the well-known control algorithms (the most popular one is the PID algorithm). The choice of the algorithm and the values of its parameters is done only once – at the stage of the control system design. The algorithm is then embedded in a hardware controller, so the control loop thus comprises of the controller, a single sensor, a plant being controlled, and a single actuator. In a modern industry such simple control loops are becoming rare, since the optimisation of the control became an important issue, due to rising costs of materials and energy. Advanced process model-based control algorithms

are now at the center of interest, due to their capabilities of adjusting a control strategy based on a forecast of the process course. In this way, it has become possible to hold the control process in a state which would not be available in the case of traditional control algorithms. For example, a plant may be set to work at the edge of the stability region or even in the controlled oscillations – wherever the efficiency of the process is better. A frequently used method for such an advanced control is the designation of certain states of the process (e.g. as a function of the process variable). The control algorithms are then changed and/or tuned depending on the state detected at the moment. The method is known under multiple names e.g. hybrid control, discrete event systems, or a multimodal control [2]. In practice, the detection of the current state of the process and choice of the corresponding control algorithm are not trivial tasks and require a complex reasoning, which is often implemented in the form of a multi-agent system (see e.g. [3]).

The control of continuous processes has to be time-determined, so the complex agent environments, designed to be flexible, not lean, are not fitted to the task. Inferring on the basis of process variables values is on the other hand well suited for blackboard-based reasoning. When the knowledge table is relatively compact (in terms of size) the timeliness of the communication may be ensured by employing a specialised real-time communication architecture in the place of a traditional server-client architecture. The implementation of the efficient communication architecture for agent-based hybrid control of industrial continuous processes is the main contribution of this work.

2 The Domain under Consideration

The proposed framework is suitable for improving the operation of distributed control systems for continuous processes in such a branches of industry as biotechnology, energy and heating, chemical, food processing. In classical distributed control systems typically three-layered model of communication is used, consisting of operating level, control level and device level. The main control tasks are serviced by the control and device level networks built usually with the specialised industrial networking instrumentation. The industrial networks (i.e. fieldbuses) are offered by various manufacturers and are based on varying architectures and protocols, so generally they are not compatible with each other. The algorithms for continuous control are mostly based on the classical PID or belong to the family of adaptive and/or predictive algorithms (see [4] and [5] for examples).

The progress in the field of process technology poses the new challenges to the process automation engineers, which cannot be answered with even the most advanced of the determined control algorithms. There are multiple classes of processes, which require additional actions, beside the ones taken by the continuous control algorithms. The additional action often require some form of advanced reasoning and/or user cooperation. As the examples of these advanced processes an activated sludge bioaugmentation [6] or a multi-phase flow control through non-horizontal piping [7] may be cited.

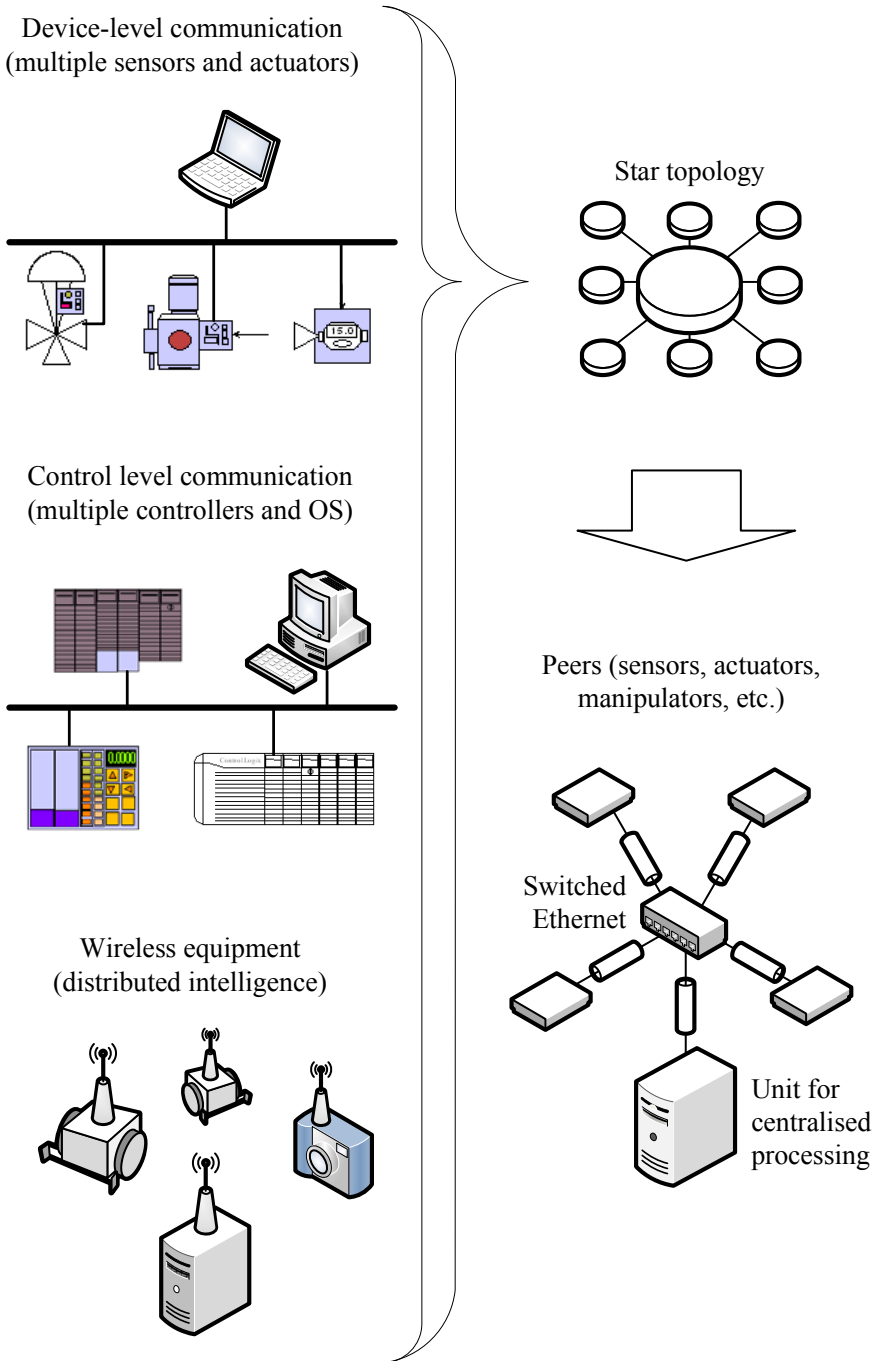


Fig. 1. Star topology in industrial process automation

The main application of the proposed communicational framework was the control of the stable oscillations in the biotechnology process simulation. In the described process the bioaugmentation is enhanced by sustaining the oscillations as it increases the lipase production in a bioreactor.

This kind of complex control requires not only new algorithms involving a combination of classical algorithms with the agent technology, but also reconfiguration of the control instrumentation and software, which bought straight in the commercial market are not fitted for the use of agent technology.

The Fig. 1 illustrates the process of transition from the standard layered networking architecture employing the commercial available instrumentation with the bus topology to the proposed pPDC architecture for star topology-based agent communication. It should be noted, that the hardware used in the proposed architecture is available as typical of-the-shelf product. Widely available devices are used such as sensors, actuators, Programmable Automation Controllers, etc. The only requirement for a device to be connectible to the pPDC framework is the compatibility with the Ethernet networking standard. This is due to the fact, that the first implementation of the pPDC is Ethernet-based. The Ethernet standard was chosen due to its popularity and tendency to unify the industrial networking standards.

3 The Architecture of pPDC-Based Agent System

The idea of software agents was particularly well adopted in the domain of complex manufacturing systems. The manufacturing systems are discrete while being distributed, which makes them a good subject of agent-based modelling and control methods [8], [9].

In the continuous process automation, agent systems are not widely used because there is no clear way of assigning the agents to the continuous physical phenomena. Moreover, the primary requirement for the control system of a continuous process is a strict time determinism. Complex agent platforms are not well fitted for fulfilling the requirement. Few approaches are found in the literature, where software agents are employed in the process automation, e.g. [10], [11], [12], but still the real-time requirement is usually left uncovered.

3.1 Blackboard Broadcasting

To present the usefulness of the pPDC communication architecture as the agent communication platform, a blackboard-based multi-agent system has yet to be defined. As [13] states “a blackboard system is characterised by two main attributes:

- a collection of independent entities known as knowledge sources, each of which has some specialised knowledge. [. . .];
- a shared data structure known as a blackboard, which knowledge sources use to communicate.

Knowledge sources in blackboard systems are capable of reading and writing to the blackboard data structure, and problem solving proceeds by the knowledge sources each monitoring the blackboard and writing to it when they can contribute partial problem solutions.”

The theoretical definition of an blackboard system gives no clues on the organisation of blackboard access attempts in order to ensure consistency of communication. This problem is solved practically in different ways, such as mechanisms of notifications of agents on the changes in blackboard [14], or scheduling a pre-defined sequence of communication between agents [15]. Overall, however, all the mechanisms are based on a classic architecture of data exchange, i.e. a client-server architecture. A detailed survey on the mechanisms of multi-agent environments is contained in [16].

In this work, to ensure determinism of the time, which is necessary in control of continuous processes, a different mechanism for broadcasting a blackboard is introduced. In place of the well-known client-server architecture, a modified scheme of the producer-distributor-consumer is employed. This architecture, in its original form, is used to organise communication in low-level industrial networks and a leading example of its use is the WorldFIP network. The individual layers of the WorldFIP standard are currently standardised as follows: WorldFIP physical layer is defined in IEC 61158 standard as separate layers, i.e. physical layer of type 1, the data link layer of type 7, and the application layer of type 7. The proper selection and combination of the individual layers of the 61158 standard in order to obtain the functional network known as WorldFIP is defined in IEC 61784 standard as a profile designated as CPF-5.

For the purposes of inter-agent communication the PDC architecture has been modified to take the link-layer mechanisms that exist in conventional computer networks into the account. This allowed the optimisation of the architecture and significantly increased its time efficiency while keeping the compatibility with the system blackboard-based inferring systems. The modified scheme has been called a parallel Producer-Distributor-Consumer and designated with the pPDC acronym. This newly invented architecture has already been implemented and tested in robotic systems [1], and the primary contribution of this work is its adaptation to the needs of agent-based control systems for continuous processes.

Each sampling period of continuous signals in the system begins with a broadcast of a blackboard content over a network that connects all the hardware platforms forming the control system. Therefore, each node receives the same unambiguous information on the system state and, by the mere fact of broadcasting, the information that another sampling period has actually started. In response to the broadcast, each of the nodes performs its internal operations involving reasoning with the freshly received information taken into the account. When the node decides that the content of the blackboard needs to be modified, it sends the modification request back to the distributor.

The general idea of the framework is illustrated in the Fig. 2. According to the terminology introduced in [16] the pPDC architecture qualifies as an interaction mediation mechanism. It provides environment-mediated interaction channel combined with centralised synchronisation mechanism in order to provide a complete solution for pervasive applications. An environment-mediated interaction channel is the form of infrastructure for broadcasted messages sent to all agents, while the specific timing of broadcasts serves as a synchronisation method for the agents implemented as parallel threads.

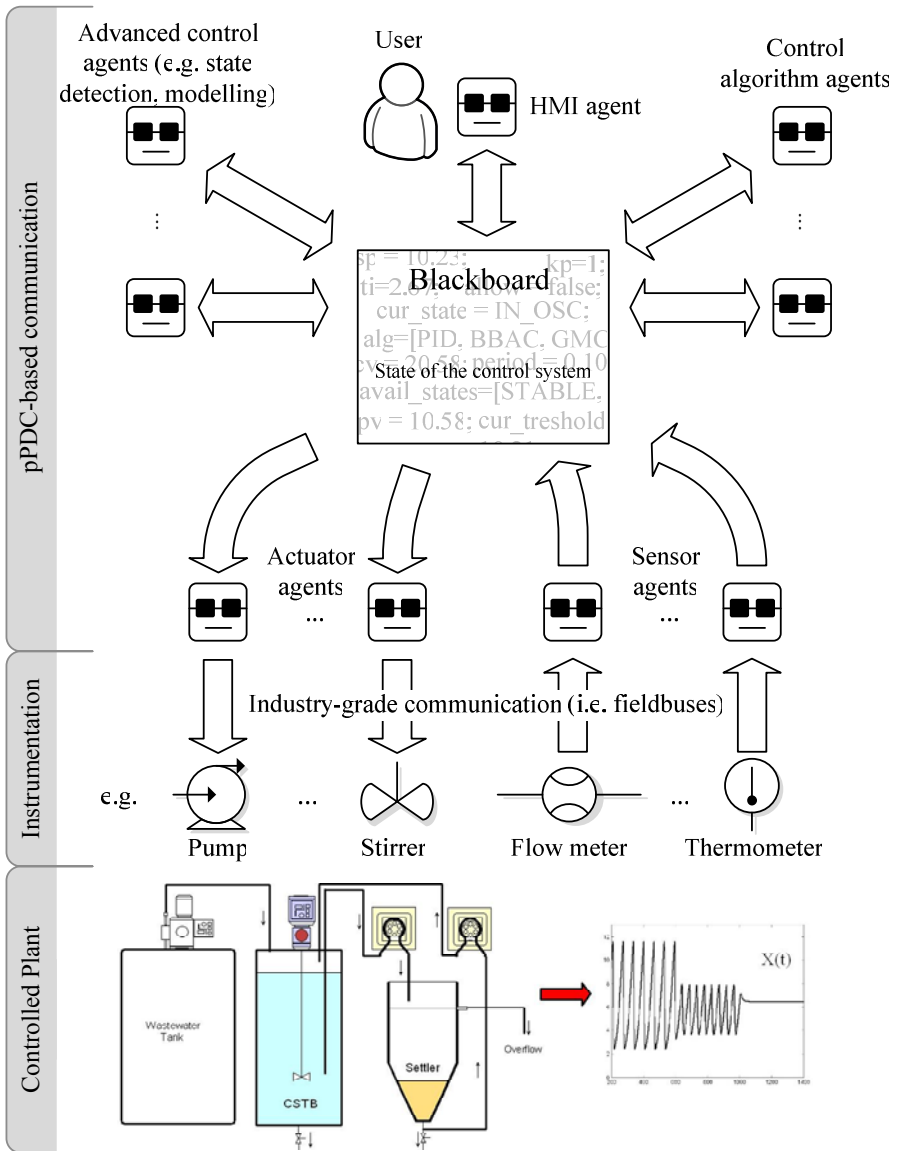


Fig. 2. The idea of agent-based advanced process control

The agents forming the system assume the roles of producers, consumers and producers/consumers, depending on their need for interaction with the blackboard. For example, Actuator Agents act as consumers, because they passively monitor the blackboard and do not attempt to modify it. On the other hand, the Sensor Agents act as pure producers, as they are not interested in the blackboard content, and they simply read values of real world signals, and include the read samples in the blackboard.

All the more complex agents act as consumers/producers, as they both react to the content of the broadcasted blackboard, and then attempt to modify it. The distributor node should not be seen as one of the agents, but rather as a part of the agent platform itself, providing the communicational interface to the blackboard.

3.2 Network Architecture and Hardware Layer

The primary assumption of the pPDC architecture is the use of cyclical broadcasts of the blackboard, therefore the networking layer chosen for the implementation has to support broadcast transmissions. Such a functionality is included in the widely used IP protocol paired with the overlaying UDP transport layer protocol. The data link layer and physical layer are proposed to be adopted directly from the IEEE802 standard. These networks, in their original form, functioned with the CSMA media access algorithm as a basis. The CSMA algorithm uses BEB mechanism, which effectively makes it impossible to designate the exact timing of network messages. However, CSMA / CD is no longer used as a media access method, since the switching technology became popular. Nowadays, virtually all cable-based networks operate in a star topology integrated by a hardware switching unit. This progress was widely noted in the literature, due to the fact, that after the elimination of the CSMA / CD mechanism the IEEE802.3 compliant networks became time determined and more reliable [17].

It is the star topology, which is natural in many process automation systems, and at the same time reflects the exact topology of blackboard-based knowledge system. In such the knowledge system, the blackboard holder is the privileged agent. All the other agents are logically connected to the blackboard only and they do not communicate with any other node. This logical structure is reflected in the hardware star topology of underlying Ethernet network. This analogy is exploited by the pPDC architecture – it is proposed to designate a network node to act as a repository of knowledge and hold the content of a blackboard. At the same time, the designated node fulfils the role of a distributor at the protocol level, and of a network switch at the link layer.

The underlying network should work on the basis of switching technology, not with the CSMA algorithm. To ensure the time determinism of the network it should work in full-duplex mode (half-duplex is not acceptable, as it results in collisions and invoking the CSMA algorithm). The most popular solution fulfilling the requirements is the 10/100/1000BASE-T Ethernet cabling. It should be noted, that all the standards of the IEEE802 family are implemented in such a way, that they are seen by the operating system as an uniform LLC layer. As a result the application layer protocols (such as the pPDC protocol) cannot determine whether the underlying network is time determined or not. On the other hand, pPDC implemented on top of non-time determined network still functions flawlessly (apart from the real-time requirement), so it still is usable for mere multi-agent integration in situations when the strict real-time performance is not required. In such cases the classical 10BASE2 cabling with the CSMA/CD method or the modern wireless networks of the IEEE 802.11 family employing the CSMA/CS method may be considered. There are also methods to ensure real-time performance in wireless networks – see [18]. When the pPDC is implemented on top of the UDP/IP protocol, the compatibility of the framework extends to the broad range of IP-enabled devices such as computers of various operating

systems, mobile phone, pads, netbooks, etc. Of course in this case the time determinism is not ensured, but the framework remains functional.

The chosen hardware platform determines the limitations of the framework:

- limit on the frequency of broadcasts – one cycle has to be long enough to fit the blackboard broadcasts and possible answers;
- limit on blackboard size – the size of broadcasted message is limited by the hardware (e.g. maximum frame length).

3.3 Performance

The sequence of events in a single cycle of the pPDC protocol is relatively simple (see Fig. 3). The cycle begins with the blackboard broadcast done by the distributor node. Preparation of the message to be sent and processing it by the protocol stack takes T_{pr} , and the physical transmission itself takes T_{tr} . The message is received and processed by all the agents connected with the framework, which perform their internal reasoning and prepare answers containing the proposed modifications to the knowledge base. These responses are then sent to the distributor by the point-to-point message with no specific coordination. This lack of coordination is not an obstacle here, because the link layer in the underlying network is reliable and it queues the messages and delivers them in an orderly fashion. The queuing and delivering the responses takes T_{que} . After receiving and interpreting all the modifications to the blackboard, the distributor waits for the next designed sampling time instant, when it begins a new cycle by broadcasting the blackboard.

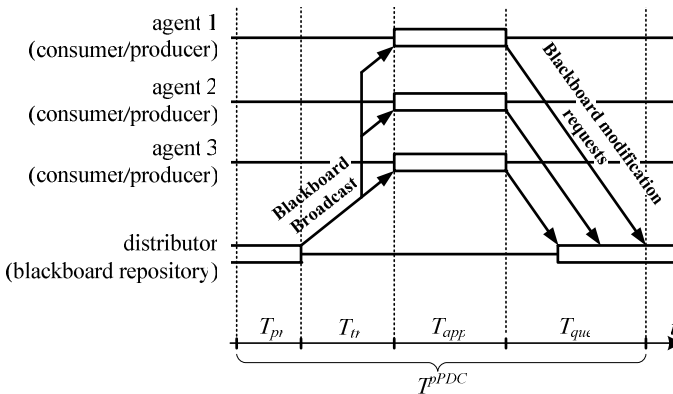


Fig. 3. Diagram of pPDC protocol interactions

The timeliness of the communication is the priority. The broadcasts of the blackboard content are evenly spaced in time (as required by methods of the continuous process control), and agents should be prepared, that the requests for the blackboard modification have to be sent during the proper time slot, or else will be ignored and overrode by the next blackboard broadcast.

It should be noted that the main advantage of the pPDC protocol, mentioned in previous publications on the subject ([1][19][20]), is that the most time-consuming

process i.e. processing of the network messages in the protocol stacks ([17]) of the distributed nodes, is performed in parallel in all the nodes. This results in a significant reduction of time needed to perform all the operations in one cycle, when compared to classical sequential protocols. It is the parallelisation of T_{app} times, which is the distinguishing feature of the pPDC architecture making it suitable for applications in time-determined continuous process control.

4 Concluding Remarks and Future Work

This work proposes a joint hardware-software environment for integrating multi-agent systems operating on the knowledge base in the form of blackboard. The protocol used in the proposed framework is an innovative solution and it allows the agent systems to operate in a time-determined manner. This in order opens the possibility of employing advanced, agent-based reasoning in the industrial control of otherwise hardly controllable processes, which are becoming more common in industrial reality.

The solution was modeled on real-time industrial networks, and complemented by the notions from the field of parallel processing, resulting in the protocol of a very high performance. The performance was evaluated in a series of experiments and proved to be nearly an order of magnitude higher than in similar protocols (see [1] for details). Currently, further plans focus on settling the protocol in practical industrial applications and research. The direction of development considered is the integration of the framework with popular widely used multi-agent and industrial holonic software solutions. Especially noteworthy are the JADE environment implementing the FIPA specification and the FBDK development environment for the IEC61499 standard.

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education under grant no. N N514 471539.

References

1. Polaków, G., Metzger, M.: ppPDC Communication Framework – A New Tool for Distributed Robotics. In: Carpin, S., Noda, I., Pagello, E., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2008. LNCS (LNAI), vol. 5325, pp. 195–206. Springer, Heidelberg (2008)
2. Faraut, G., Pietrac, L., Niel, E.: Formal Approach to Multimodal Control Design: Application to Mode Switching. *IEEE T. Ind. Inform.* 5, 443–453 (2009)
3. Machado, V., Neto, A., de Melo, J.: A Neural Network Multiagent Architecture Applied to Industrial Networks for Dynamic Allocation of Control Strategies Using Standard Function Blocks. *IEEE T. Ind. Electron.* 57, 1823–1834 (2010)
4. Czczot, J., Laszczyk, P., Metzger, M.: Local balance-based adaptive control in the heat distribution system - Practical validation. *Appl. Therm. Eng.* 30(8-9), 879–891 (2010)
5. Nocoń, W., Metzger, M.: Predictive Control of Decantation in Batch Sedimentation Process. *AICHE J.* 56(12), 3279–3283 (2010)
6. Turek-Szytow, J., Choinski, D., Miksch, K.: Properties of the activated sludge after lipase bioaugmentation. *Environ. Prot. Eng.* 33, 211–219 (2007)

7. Choiński, D., Metzger, M., Nocoń, W.: Multiscale three-phase flow simulation dedicated to model based control. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part II. LNCS, vol. 5102, pp. 261–270. Springer, Heidelberg (2008)
8. Van Dyke Parunak, H., Savit, R., Riolo, R.L.: Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In: Sichman, J.S., Conte, R., Gilbert, N. (eds.) MABS 1998. LNCS (LNAI), vol. 1534, pp. 10–25. Springer, Heidelberg (1998)
9. Jennings, N.R., Bussmann, S.: Agent-based control systems. *IEEE Contr. Syst. Mag.* 23(3), 61–73 (2003)
10. Van Breemen, A.J.N., De Vries, T.J.A.: Design and implementation of a room thermostat using an agent-based approach. *Control Eng. Pract.* 9(3), 233–248 (2001)
11. Seilonen, I., Koskinen, K., Pirttioja, T., Appelqvist, P., Halme, A.: Reactive and deliberative control and cooperation in multi-agent system based process automation. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 469–474. IEEE Press, New York (2005)
12. Guo, Y.-N., Cheng, J., Gong, D.-w., Zhang, J.-h.: A Novel Multi-agent Based Complex Process Control System and Its Application. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCIS, vol. 344, pp. 319–330. Springer, Heidelberg (2006)
13. Wooldridge, M.: *An Introduction to MultiAgent Systems*. Wiley, Chichester (2002); ISBN 0-71-9691-X
14. Li, S., Hu, F.: Communication between the RoboCup Agents Based on the Blackboard Model and Observer Pattern. In: 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–5. IEEE Press, New York (2009)
15. Lieping, Z., Yunsheng, Z.: Research on Hierarchical Distributed Coordination Control in Process Industry Based on Multi-agent System. In: 2010 International Conference on Measuring Technology and Mechatronics Automation, pp. 96–100. IEEE Computer Society, Los Alamitos (2010)
16. Platon, E., Mamei, M., Sabouret, N., Honiden, S., Parunak, H.V.D.: Mechanisms for environments in multi-agent systems: Survey and opportunities. *Auton. Agent. Multi-Ag.* 14, 31–47 (2006)
17. Lee, K.C., Lee, S.: Performance evaluation of switched Ethernet for real-time industrial communications. *Comp. Stand. Inter.* 24, 411–423 (2002)
18. Yoo, S.-e., Chong, P.K., Kim, D., Doh, Y., Pham, M.-L., Choi, E., Huh, J.: Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4. *IEEE T. on Ind. Electron.* 57, 3868–3876 (2010)
19. Polaków, G., Metzger, M.: Agent-Based Approach for LabVIEW Developed Distributed Control Systems. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 21–30. Springer, Heidelberg (2007)
20. Polaków, G., Metzger, M.: Agent-Based Framework for Distributed Real-Time Simulation of Dynamical Systems. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, pp. 213–222. Springer, Heidelberg (2009)

Sequence Automata for Researching Consensus Levels

Henryk Piech

Czestochowa University of Technology,
Dabrowskiego 73

h.piech@adm.pcz.czest.pl

Abstract. Having a group of ranking lists (ordered sequences) we can use different theories to research a scale of consistency according to the assessments contained in it. The agreement of ordered (queue) objects can be estimated as the maximal length of preference subsequences i.e. a maximal set of objects with the same queue of dominance relation. It is not important that among the chosen objects others exist but it is important that the i -th object is before j -th objects (has greater validation). The proposed sequence automata can be use to find different kinds of sequence agreements or disagreement among a sequenced group represented by for example by a ranking list. An exploitation ranking list has an essential advantage over other methods of object judgment (assessments) because different criteria or their sets are reduced only to positions on lists. In such a situation the problem of assessing the scale of consistency (agreement among authors or algorithms of ranking lists) remains. A unified-form of complex criteria presentation helps us to elaborate the tools for consistency estimation. We expect that sequence automata will resolve this and similar problems connected with sequence analysis. Our proposition is the tool supporting multi-agents system in areas of negotiation, partner recognition, creation hierarchy of criteria, planning strategy and finding hidden, imperceptible nuances (details), having dependence (or preference) character. It can be also use for coding and decoding information by sequence interlacement.

Keywords: discrete optimization, ranking lists, consistency estimation.

1 Introduction

Sequence features are exploited in many discrete optimization tasks such defining critical path, shortest road, maximal flow, cardinality matching, minimal spanning tree, scheduling tasks etc. [1,2,4,6,7,14,21]. In some problems global characteristics of sequences (shortest path, chipset flow) are used in others one of the chosen (most important) properties (capacity in maximum flow) [9]. There are also tasks where particular relations are defined between sequence elements [3,5,8], or mutual relations among a set of assessed elements. Sometimes they are regarded as topological features of the respective adjacency of elements

(cardinality matching). In a group of problems elements are gathered in a class or category sets. In this case we simultaneously gain and lose part of the information (connected with the group and individual mutual relations which can be expressed by entropy level [22]). In the problem of scheduling, and timetabling, ordered sequences are additionally exploited as initial structuralized data [18,21]. The same problems are directly connected with individual mutual relations among elements (ranking lists) and expose the task of final objects placing [15,19] (ranking lists). Our approach is strictly connected with the treatment of such sequences. More over, we want to evaluate the scale of sequence consistency in a set of ranking lists. Sequence consistency can be researched with the help of preference relations, neighborhood relations, likelihoods, and pairwise relations theories [16,25]. We can adapt the sequences order approach to asses both objects and experts or algorithms generating a ranking list for concrete objects sets. Sequence analysis concerns the problems of order objects agreement (disagreement) presented by subsequences, neutral elements which are not included in maximal length agreement (disagreement) sets or neutral elements which are included in agreement (disagreements) sets but not with maximal length. To make such a type of analysis we propose to exploit sequence automata and present a system of coding states and states changing functions. There is possibility of realizing operations on sequences and defining an operations assortment set. The idea can be use for agent's communication and decision support system as well as for other support systems that are not connected, or only indirectly connected, with agents activities e.g. psychology, genetics, sport support systems, etc. Multi-agent systems based on sequence recognition fulfill the strategies of:

- negotiating, where we should regard preferences, domination of factors, events, attributes, declarations etc., there, the scale of agreement or disagreement among negotiators can be estimated [26,29],
- creating cooperation agent's groups according to a given stencil, describing organization's abilities, predispositions, strategies, mental abilities and resources, knowledge and skills [27,31],
- marketing and games, regarding scale, location and chronology of actions [32],
- finding partners based on ordered features of characters (e.g. in form of preference list or model) [30,33], Using sequences automata we can find not only closed sequences but also hidden interlaced sequence structures, e.g. in:
 - finding DNA chromosome structure (gene sequences) [28],
 - recognizing psychological spectrums created on a base of validation hierarchy [31].

The paper consists of 5 sections. Section 2 is devoted to the presentation of sequence automata grammar. In section 3 a graph structure of automata and its nodes is presented. A system of node states and state changes functions coding is proposed in section 4.

2 Sequence Automata Grammar

In the proposed automata we can identify several main elements. The first is the alphabet of accepted and unaccepted states $\Sigma(\Sigma_a \cup \Sigma_u)$. Next we introduce the obvious initial state $(startS_1, startS_2, \dots, startS_{ls}) = \sigma_0$, where ls - the number of sequences (minimum two compared sequences). In the set of sequences we designate stencil one. The initial state refers to the all compared sequences. We usually initiate the work of sequence automata from their initial positions. It is convenient to consider only two sequences and then expand it to several others. Additionally it should be enriched with states changing formulas $f_i \in F, i = 1, \dots, lf$. Our automata consist of sequences in which we want to find the consistency of queuing order. It is, in fact, hidden consistency that consists in the maximal length sequences (optimal agreement) of the same objects, among them can be other randomly located objects in the compared based sequence S_1 (stencil) and the remaining S_2, S_3, \dots . When searching for analogies in common life we want to find, define or estimate the scale of agreement between two (or several) persons, which is imperceptible at first glance. Therefore sequence automata can be defined by 4-tuple $\{\Sigma, \sigma_0, F, \sigma\}$. Let us consider example of sequence automata application based on searching for a maximal length elements (objects) queue in two sequences. The functioning of sequence automata can be described with the help of the diagrams in fig.1. The full cycle of sequence automata work consists of n episodes. For a stencil sequence, states changes are realized from the starting position to the last acceptable position, after that the starting position is modified by increment (function f_4) and we return to the new starting position $startS_1$. Thus, every episode starts from the next states

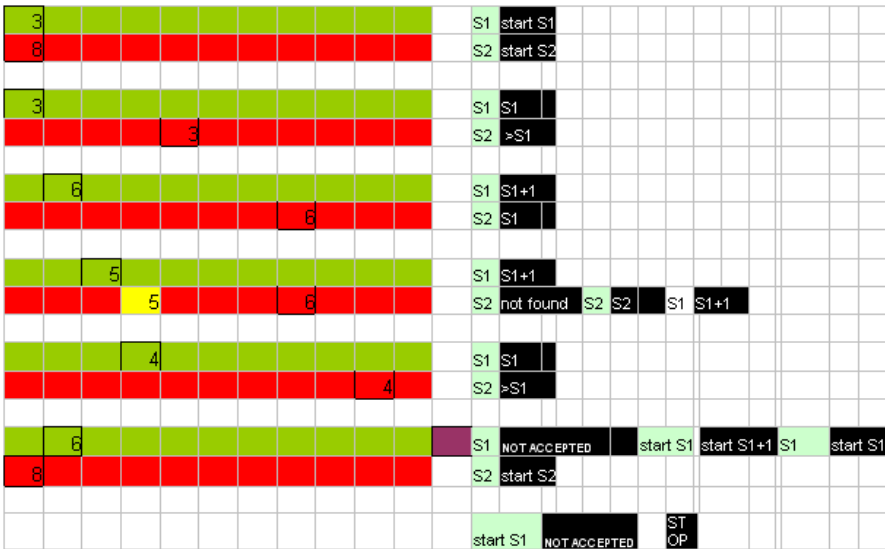


Fig. 1. Example of finding hidden sequences agreement with help of sequence automata

$startS_1 = j, j = 1, \dots, n$ for first sequence and from the same states $startS_2 = 1$ for the second sequence (for simplification we assume that the length of all sequences are equal). The procedures of changing automata states presented in atomic form can be described as follows:

- $f_1 = \{\sigma_{S_1} = startS_1, \sigma_{S_2} = startS_2\}$
- $f_2 = \{\sigma_{S_2} = \sigma_{S_1}$ if object σ_{S_1} is found in right hand side of second sequence from actual state σ_{S_2} otherwise $\sigma_{S_2} = \sigma_{S_1}\}$
- $f_3 = \{\sigma_{S_1} = \sigma_{S_1} + 1$ and if σ_{S_1} is accepted come back to $f_2\}$ (*end of sub-episode*)
- $f_4 = \{startS_1 = startS_1 + 1$ if new $startS_1$ is accepted (not greater than length of first sequence) otherwise *STOP*\}(*end of episode*)

To complete our example-task we should entail counting number of situations when the same object is found in compared sequence $\sigma_{S_2} = \sigma_{S_1} : l_2 = l_2 + 1$. At the same time we cannot forget about resetting counter $l_2 = 0$, each time we change the state of the first sequence: $f_3 = \{\sigma_{S_1} = \sigma_{S_1} + 1\}$ or when we start a new sub-episode: $f_4 = \{startS_1 = startS_{1+1}\}$ and then $f_1 = \{\sigma_{S_1} = startS_1, \sigma_{S_2} = startS_2\}$. Before resetting counter l_2 , the actual and maximal queue must be saved :

- $n_act; obj(\sigma_{S_2}^{(1)}), obj(\sigma_{S_2}^{(2)}), \dots, obj(\sigma_{S_2}^{(n_act)})$,
 - $n_max; obj(\sigma_{S_2}^{(1)}), obj(\sigma_{S_2}^{(2)}), \dots, obj(\sigma_{S_2}^{(n_max)})$,
- where n_act, n_max - are respectively the length of the actual agreement queue and length of maximal agreement queue.

An agreement queue is connected with one particular episode. The episode can be realized as follows: $f_1, f_2^{(1)}, f_2^{(2)}, \dots, f_2^{(n_act)}, f_3$

3 Proposition of Graph Presentation of Sequence Automata

Let us start from recalling that states changing is connected with changing the stencil (*ss*) and compared (*sc*) sequences position (Fig.2). Hence the number of possible sequence automata states can be estimated as:

$$nsa = \sum_{j=1}^m n(i) * (n(i) + 3)/2$$

ss			
sc1	sc2	sc(m-1)

Fig. 2. Information for single node equivalent concrete state in sequence automata

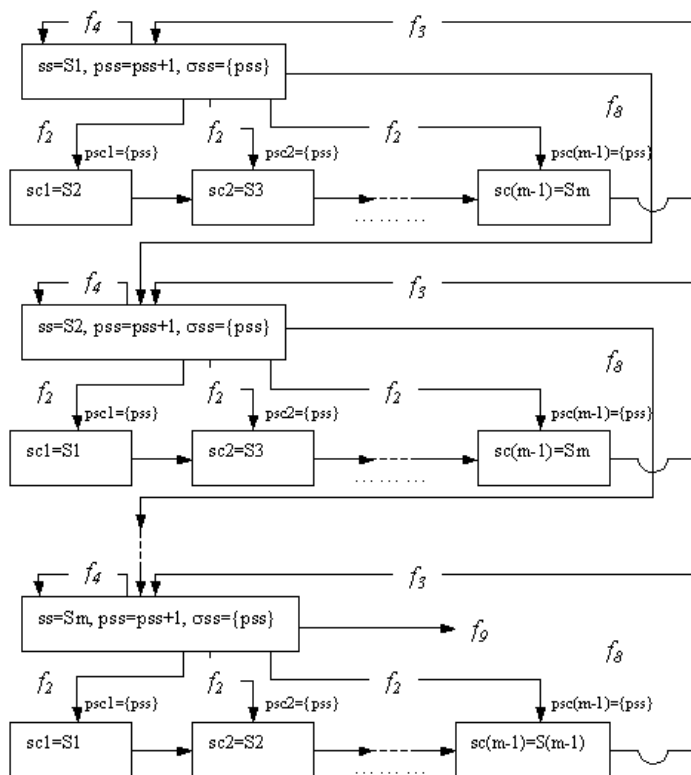


Fig. 3. Scheme of sequence automata functioning - example, here $f_8 = \{ss = ss + 1, sc1 \neq ss, sc2 \neq ss, \dots, sc(m-1) \neq ss, sci \neq scj\}$, ss =stencil sequence, sci - i -th compared sequence, $f_9 = \{Stop\}$

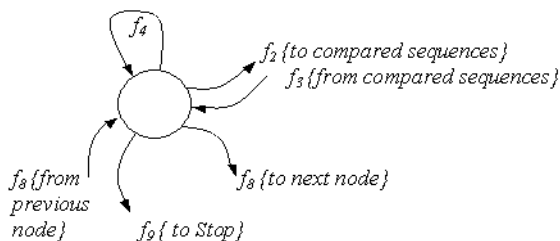


Fig. 4. Figure of sequence automata node which can have $n_1 + n_2 + \dots + n_m$ states

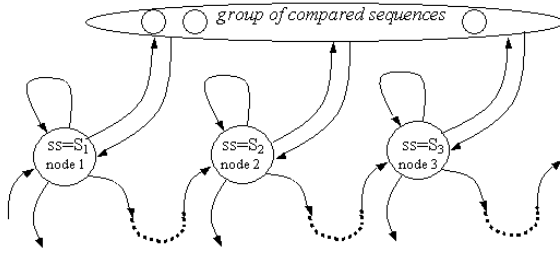


Fig. 5. Form of sequence automata nodes connections

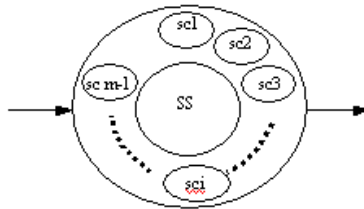


Fig. 6. Node as a state of sequence automata

where $n(i)$ - length of i -th sequence By including functions of states transformation we can describe graphically the functioning automata process(Fig.3). The general figure of sequence automata node is depicted in Fig.4. The method of nodes connection is presented in Fig.5. If we want to retain adequate nodes and states we can use another form of nodes presentation (Fig.6)

4 System of States and Functions Coding

It is proposed to exploit vectors coding according the following structure organization:

$$[code_{ss} | code_{sc}(1), \dots, code_{sc}(i - 1), "0" - insted\ of\ sc(i) = ss, code_{sc}(i + 1), \dots, code_{sc}(m)]$$

Sequence ss was designated as stencil and its position in the vector was fulfilled by "0" . The start state of the node will be defined as $[1|0, 1, , 1]$, where before " | " we have only one vector element (adequated position in a stencil sequence) and after " | " we have m vector elements (adequated positions in compared sequences and "0" in an empty place representing a removed stencil sequence. We decide to describe example, which can be presented with the help of the set of vectors:

1st meta-episode

<i>episode1</i>	<i>episode2</i>	<i>episode3</i>
[1 0, 1, 1, 1]	[2 0, 1, 1, 1]	[3 0, 1, 1, 1]
[1 0, 4, 5, 3] $l2 = 1$	[2 0, 3, 3, 4] $l2 = 1$	[3 0, 1, 2, 6] $l2 = 1$
[2 0, 4, 5, 3]	[3 0, 4, 5, 3]	[4 0, 1, 2, 6]
[3 0, 4, 5, 3]	[4 0, 4, 5, 3]	[5 0, 1, 2, 6]
[4 0, 4, 5, 3]	[5 0, 4, 5, 3]	[6 0, 1, 2, 6]
[5 0, 4, 5, 3]	[6 0, 6, 6, 5] $l2 = 2$	
[6 0, 6, 6, 5] $l2 = 2$		

<i>episode4</i>	<i>episode5</i>	<i>episode6</i>
[4 0, 1, 1, 1]	[5 0, 1, 1, 1]	[6 0, 1, 1, 1]
[4 0, 2, 4, 1] $l2 = 1$	[5 0, 5, 1, 2] $l2 = 1$	[6 0, 6, 6, 5] $l2 = 1$
[5 0, 2, 4, 1]	[6 0, 6, 6, 5]	
[6 0, 6, 6, 5] $l2 = 2$		

4th meta-episode

<i>episode1</i>	<i>episode2</i>	<i>episode3</i>
[1 1, 1, 1, 0]	[2 1, 1, 1, 0]	[3 1, 1, 1, 0]
[1 4, 2, 4, 0] $l2 = 1$	[2 5, 5, 1, 0] $l2 = 1$	[3 1, 4, 5, 0] $l2 = 1$
[2 4, 2, 4, 0]	[3 5, 5, 1, 0]	[4 1, 4, 5, 0]
[3 4, 2, 4, 0]	[4 5, 5, 1, 0]	[5 6, 6, 6, 0] $l2 = 2$
[4 4, 2, 4, 0]	[5 6, 6, 6, 0] $l2 = 2$	[6 6, 6, 6, 0]
[5 6, 6, 6, 0] $l2 = 2$	[6 6, 6, 6, 0]	
[6 6, 6, 6, 0]		

<i>episode4</i>	<i>episode5</i>	<i>episode6</i>
[4 1, 1, 1, 0]	[5 1, 1, 1, 0]	[6 1, 1, 1, 0]
[4 2, 3, 3, 0] $l2 = 1$	[5 6, 6, 6, 0] $l2 = 1$	[6 3, 1, 2, 0] $l2 = 1$
[5 6, 6, 6, 0] $l2 = 2$	[6 6, 6, 6, 0]	
[6 6, 6, 6, 0]		

The maximum agreement queue is equal to $\max\{l2\} = 2$ (in the presentation we omit the 2nd and 3rd meta-episode but it does not introduce any changes). Let us test the scale of disagreement queue.

1-st meta-episode

<i>episode1</i>	<i>episode2</i>	<i>episode3</i>
[6 0, 1, 1, 1]	[5 0, 1, 1, 1]	[4 0, 1, 1, 1]
[6 0, 6, 6, 5] $l3 = 1$	[5 0, 5, 1, 2] $l3 = 1$	[4 0, 2, 4, 1] $l3 = 1$
[5 0, 6, 6, 5]	[4 0, 5, 1, 2]	[3 0, 2, 4, 1]
[4 0, 6, 6, 5]	[3 0, 5, 1, 2]	[2 0, 2, 4, 1]
[3 0, 6, 6, 5]	[2 0, 5, 1, 2]	[1 0, 2, 4, 1]
[2 0, 6, 6, 5]	[1 0, 5, 1, 2]	
[1 0, 6, 6, 5]		

<i>episode4</i>		<i>episode5</i>		<i>episode6</i>
[3 0, 1, 1, 1]		[2 0, 1, 1, 1]		[1 0, 1, 1, 1]
[3 0, 1, 2, 6]	<i>l3</i> = 1	[2 0, 3, 3, 4]	<i>l3</i> = 1	[1 0, 4, 5, 3]
[2 0, 1, 2, 6]		[1 0, 3, 3, 4]		
[1 0, 1, 2, 6]				

4-th meta-episode

<i>episode1</i>		<i>episode2</i>		<i>episode3</i>
[6 1, 1, 1, 0]		[5 1, 1, 1, 0]		[4 1, 1, 1, 0]
[6 3, 1, 2, 0]	<i>l3</i> = 1	[5 6, 6, 6, 0]	<i>l3</i> = 1	[4 2, 3, 3, 0]
[5 6, 6, 6, 0]	<i>l3</i> = 2	[4 6, 6, 6, 0]		[3 2, 3, 3, 0]
[4 6, 6, 6, 0]		[3 6, 6, 6, 0]		[2 2, 3, 3, 0]
[3 6, 6, 6, 0]		[2 6, 6, 6, 0]		[1 2, 3, 3, 0]
[2 6, 6, 6, 0]		[1 6, 6, 6, 0]		
[1 6, 6, 6, 0]				

<i>episode4</i>		<i>episode5</i>		<i>episode6</i>
[3 1, 1, 1, 0]		[2 1, 1, 1, 0]		[1 1, 1, 1, 0]
[3 1, 4, 5, 0]	<i>l3</i> = 1	[2 5, 5, 1, 0]	<i>l3</i> = 1	[1 4, 2, 4, 0]
[2 1, 4, 5, 0]		[1 5, 5, 1, 0]		
[1 1, 4, 5, 0]				

The scale of disagreement queue is equal to $max\{l3\} = 2$ (in presentation we omit the 2nd and 3rd meta-episode but it does not introduce any changes).

Next we code the function of stages changing. This task is simple and we can code it in decimal or binary form. We can also use vector structure e.g. function f_1 we present as $[1, 0, 0, 0, 0, 0, 0, 0, 0]$ or simply $[10000000]$ (presently there is a defined set of f_i). It is a convenient form because we can combine several function s in one vector. Therefore the sequence of functions may be presented in matrix form in which every row contains a code of functions in the following time periods. For the above presented task we use the series of functions: $f_1, f_7, \dots, f_7, f_3, f_4$ and it put on matrix form: vectors are adequated matrix rows. Obviously by increasing the number of functions the number of columns will be increased as well. In different cases, not only the scale of agreement or disagreement is interesting, but also the set of elements which create agreement (or disagreement) subsequences:

$$\bigcup_{k=1}^{l-n-max} obj^k(\sigma_{SC}^{(1)}), obj^k(\sigma_{SC}^{(2)}), \dots, obj^k(\sigma_{SC}^{(n-max)}),$$

where

$l-n-max$ - the number of maximal length agreement (disagreement) subsequences,
 $obj^k(\sigma_{SC}^{(i)})$ - code of i -th object in k -th subsequence.

The matrix form of functions coding permits us to describe several functions in one vector and designate them to chronological realization according to code "1" placements from the left hand side.

5 Conclusions

Sequence automata can be exploited for realization operations and a complex task over a set of sequences. Searching for maximal length queue agreement (or disagreement) is an example of a task using a set of sequences. Operations employing sequences help assess consistency among task,object lists.

Unaccepted states in compared sequences generally leads to establishing initial state in stencil sequence. Not accepted states in stencil sequence carries on to task finish.

References

1. Ahuja, R., Magnanti, T., Orlin, J.B.: Network Flows. Prencice-Hall, Englewood Clifs (1993)
2. Baker, K.R., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constrains, TR 8028/0, Econometric Institute, Erasmus University, Rotterdam (1980)
3. Baker, K.R., Schrage, L.E.: Finding an optimal sequence by dynamic programming: An extension to precedence-related tasks. *Oper. Res.* 26, 111–120 (1978)
4. Ballas, E.: Machine sequencing via disjunctive graphs: An implicit enumeration algorithm. *Oper. Res.* 17, 941–957 (1969)
5. Bennington, G.E., Mc Ginnis, L.F.: A critique of project planning with constrained resources. *Elmaghraby*, 1-28 (1973)
6. Blazewicz, J., Lenstra, J.K., Rinnooy Kan, A.H.G.: Scheduling subject to resource constrains: Classification and complexity. *Discrete Appl. Math.* 5, 11–24 (1983)
7. Cheriton, D., Tarian, R.E.: Finding minimum spanning trees. *SIAM J. Comput.* 5, 724–742 (1976)
8. Coffman, E.G., Graham, R.L.: Optimal scheduling for processor system. *Acta Inform.* 1, 200–213 (1972)
9. Denardo, E.V., Fox, B.L.: Shortest - route methods: 1. Reaching, purring and buckeds. *Oper. Res.* 27, 215–248 (1979)
10. Derigs, U., Heske, A.: A computational study on some methods for solving the cardinality matching problem. *Angew. Inform Jg* 22, 249–254 (1980)
11. Fredman, M.L.: New bounds on the complexity of the shortest path problem. *SIAM J. Comput.* 5, 83–89 (1976)
12. Gabow, H.: An efficient implementation of Edmond’s algorithm for maximum matching on graphs. *J. AMC* 23, 221–234 (1975)
13. Hu, T.C.: Parallel sequencing and assembly line problems. *Oper. Res.* 9, 841–848 (1961)
14. Jackson, J.R.: An extension of Johnson’s results on job lot scheduling. *Naval Res. Logist. Quart.* 3, 201–203 (1956)
15. Jackson, J.R.: Scheduling a production line to minimize maximum tardiness, RR 43, Management Science Research Project. University of California, Los Angeles (1955)
16. Keeney, R., Raiffa, H.: Decision with Multiple Objectives: Preferences and value Tradeoffs. John Wiley and Sons, NewYork (1976)
17. Kerschenbaum, A., Van Slyke, R.: Computing minimum spanning trees efficiently w. In: Proc. 25-th, Ann. Conf. of the ACM, pp. 518–527 (1972)

18. Lawler, E.L., Lenstra, J.K., Rinooy Kan, A.G.H., Shmoys, D.: The Traveling Salesman Problem. Wiley, New York (1985)
19. McNaughton, R.: Scheduling with deadlines and loss functions. *Management Sci.* 6, 1–12 (1959)
20. Neuman von, J.: Theory of Self - Reproducing Automata. University of Illinois Press, US (1966)
21. Piech, H., Ptak, A.: Use of a fuzzy strategy to create the shortest Hamiltonian cycle. In: Intern.Conf. on Fuzzy Sets and Soft Comp. in Economics and Finance, pp. 299–307 (2004)
22. Piech, H.: Scheduling with applying rough sets theory. *Czestochowa University of technology, Monographs* 149 (2008)
23. Rinnoy Kan, A.H.G., Lageweg, B.J., Lenstra, J.K.: Minimizing total cost in one machine scheduling. *Oper. Res.* 23, 908–927 (1975)
24. Robertson, T., Wright, F.T., Dykstra, R.L.: Order restricted Statistical Inference. John Wiley and Sons, NewYork (1998)
25. Roy, B.: Multicriteria Methodology for decision Aiding. Kluwer Academic Publishers, Dordrecht (1996)
26. Rosenschein, J., Zlotkin, G.: Task oriented domains: Designing Convention for Automated Negotiation among Computers, pp. 29–52. MIT Press, Cambridge (1994)
27. Schelling, T.: The strategy of Conflict, pp. 54–80. Oxford University Press, New York (1963)
28. Subrahmanian, V.S., Bonatti, P., Dix, J.: Heterogeneous Agent Systems. MIT Press, Cambridge (2000)
29. Tu, M.T., Griffel, F., Merz, M., Lamersdorf, W.: A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents. In: Rothermel, K., Hohl, F. (eds.) MA 1998. LNCS, vol. 1477, pp. 222–236. Springer, Heidelberg (1998)
30. Vasudevan, S., Kurose, J., Towsley, D.: Design and Analysis of a Leader Election. In: 12-th IEEE International Conference of Network Protocols, ICNP, Los Alamitos, pp. 350–360 (2004)
31. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley and Sons, Chichester (2002)
32. Zeleznikow, J., Bellucci, E.: Family - Winner: Integrating game theory and heuristic to provide negotiation support. In: 16-th International Conference on Legal Knowledge Based System, Amsterdam, pp. 21–30 (2003)
33. Ziegler, C.N., Lausen, G.: Propagation models for trust and distrust in social networks. *Information System Frontiers* 7, 337–358 (2005)

Agent-Based Simulation for Equilibrium Selection and Coordination Failure in Minimum Strategy Coordination Games

Ichiro Nishizaki, Tomohiro Hayashida, and Noriyuki Hara

Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan
{nisizaki,hayashida,nhara}@hiroshima-u.ac.jp

Abstract. In this paper, to analyze equilibrium selection and coordination failure in coordination games, we develop an agent-based simulation system in which artificial adaptive agents have a mechanism of decision making and learning based on neural networks and genetic algorithms. Using the simulation system, we examine the strategy choices of agents and formation of equilibria in the steady state, and compare the experimental result given by [Van Huyck *et al.* \(1990\)](#) with our simulation result.

Keywords: Minimum strategy coordination games, equilibrium selection, agent-based simulation.

1 Introduction

In this paper, we deal with coordination games with multiple equilibria which are strictly Pareto ranked. Although we can think about persuasive criteria for equilibrium selection such as payoff dominance, risk dominance and security, it is difficult to predict which equilibrium will be realized because of uncertainty about actions of opponents. Because it is difficult to deductively predict actions of players for a given game, criteria for equilibrium selection have been evaluated by experimental researches ([Cooper *et al.* \(1990\)](#); [Ochs \(1990\)](#); [Van Huyck *et al.* \(1990, 1991\)](#); [Straub \(1995\)](#); [Friedman \(1996\)](#); [Battalio *et al.* \(2001\)](#); [Morgan and Sefton \(2002\)](#); [Schmidt *et al.* \(2003\)](#); [Goeree and Holt \(2005\)](#)).

In this paper, we focus on the coordination games characterized by the minimum strategy used in the experimental research by [Van Huyck *et al.* \(1990\)](#). We develop a simulation system with artificial adaptive agents, and analyze data from the simulations with several settings. In our simulation system for the coordination games, an action of an agent is determined by a vector of outputs from a nonlinear function with several input data that the agent can know after playing a stage game. This decision mechanism is implemented by a neural network. The synaptic weights and thresholds characterizing the neural network are revised so as to obtain larger payoffs through a genetic algorithm, and then this learning mechanism develops agents with better performance.

Using the simulation system with adaptive artificial agents, we compare the result of the experiment by [Van Huyck *et al.* \(1990\)](#) with that of the simulations, and examine behavior of agents in the coordination games. Varying the degree of security of the

game, information given to players, and the number of players as parameters, we arrange three settings of the simulations. In section 2, we briefly review the experimental results of the coordination games. Section 3 is devoted to describing the agent-based simulation system with decision and learning mechanisms based on neural networks and genetic algorithms. In section 4, we examine the results of the simulations, and compare the experimental data with them. Finally in section 5, we give a summary of the simulations and some concluding remarks.

2 Minimum Strategy Coordination Game

In this section, we summarize the results of the experimental investigation by [Van Huyck *et al.* \(1990\)](#). Because the coordination games treated in the experiment are characterized by the minimum values of the strategies selected by players, we refer to this game as the minimum strategy coordination game.

Let the set of players be $N = \{1, \dots, n\}$. All the players have the common set of strategies: $S = \{1, \dots, \bar{s}\}$. Let $x_i \in S$ denote a strategy of player i . Then, the payoff function of player i is represented by

$$\begin{aligned} \pi(x_i, \underline{x}_i) &= a \min(x_i, \underline{x}_i) - bx_i + c, \\ \underline{x}_i &= \min(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n), \quad a > b > 0, \quad c > 0. \end{aligned} \quad (1)$$

The payoff of player i decreases with a strategy x_i of self, and increases with the minimum \underline{x}_i among strategies of the others. To guarantee positive payoffs, the constant c is added.

A best response to \underline{x}_i is the same as the strategy \underline{x}_i . If all the players choose the same strategy $x \in S$, the strategy profile (x, \dots, x) is a Nash equilibrium; each of the strategies $x \in S$ can potentially compose a Nash equilibrium.

When a player fails to predict the minimum strategy \underline{x}_i , it follows that the player chooses another strategy $x_i \neq \underline{x}_i$, and an outcome of the game results in disequilibrium. When multiple equilibria can be Pareto ranked, the Pareto dominant equilibrium may not always be chosen even if players can respond optimally, and then players fail to coordinate their strategies. It is supposed that an occurrence of disequilibrium or coordination failure is the most crucial issue in the coordination games.

In the minimum strategy coordination game, the strategy profile $(\bar{s}, \dots, \bar{s})$, in which all the players choose the maximum strategy \bar{s} and they obtain the maximal payoff $\bar{s}(a-b) + c$, is the payoff dominant equilibrium. Then, the maximum strategy \bar{s} is called the payoff dominant strategy. However, as the number of players n grows large, an incentive to deviate the payoff dominant strategy becomes also large because of strategic uncertainty.

The concept of security contrasts with the efficiency of the payoff dominance in the minimum strategy coordination game, and the conflict between them should be considered. The maximin strategy guarantees a certain constant payoff, and it leads to a Nash equilibrium. However, this Nash equilibrium is not efficient obviously. A player who chooses the secure strategy 1 obtains the payoff $a - b + c$, no matter which strategy the other players select; therefore the strategy profile $(1, \dots, 1)$ is called the secure equilibrium.

3 Simulation Model

In this paper, we develop a simulation system with adaptive artificial agents repeatedly playing the minimum strategy coordination games, and we analyze data from the simulations in which a variety of runs are performed by varying values of some parameters characterizing the game. An artificial adaptive agent in our agent-based simulation system has a mechanism of decision making and learning based on neural networks and genetic algorithms.

Agents repeatedly play the minimum strategy coordination game; agents obtaining larger payoff are likely to reproduce in the next period, and conversely agents obtaining only a little payoff are likely to be weeded out. In our artificial genetic system, the whole population consists of m game groups, and in each game group the minimum strategy coordination game is played by n agents.

An agent corresponds to a neural network which is characterized by synaptic weights between two nodes in the neural network and thresholds which are parameters in the output function of nodes. Because a structure of neural networks is determined by the number of layers and the number of nodes in each layer, an agent is prescribed by the fixed number of parameters if these numbers are fixed. In our model, we form a string compound of these parameters which is identified with an agent, and the string is treated as a chromosome in an artificial genetic system.

A strategy of an agent is determined by the outputs of the agent's neural network. Namely, the outputs out_s , $s = 1, \dots, \bar{s} (= 7)$ of the neural network correspond to from the strategy 1 to the strategy 7, and the strategy s^* with the largest output out_{s^*} is chosen as the next strategy of the agent.

Inputs of the neural network are the following six values, and the additional two sets of inputs, input sets A1 and A2, are used in the simulation with respect to information given to players. Each of inputs is normalized in $[0, 1]$. In the following inputs, the subscript i , $i = 1, \dots, n$ means the player i and the subscript j , $j = 1, \dots, m$ means the game group j . Thus, the subscript ij identifies a particular agent in the artificial genetic system.

[Input 1, x_{ij}] the strategy chosen by agent ij at the last period.

[Input 2, y_j] the minimum among the strategies chosen by all the agents in game group j at the last period.

[Input 3, π_{ij}] the payoff obtained by agent ij in the last period.

[Input 4, x_{ij}^T] the weighted most frequent strategy in the last T periods: $x_{ij}^T = \arg_{s \in \{1, \dots, 7\}} \max \sum_{u=t-T+1}^t e_s^u w^{u-t}$, where $e_s^t = 1$ if agent ij chooses the strategy s at period t , otherwise $e_s^t = 0$, and w is the discount factor.

[Input 5, y_j^T] the weighted most frequent minimum strategy in the last T periods: $y_j^T = \arg_{s \in \{1, \dots, 7\}} \max \sum_{u=t-T+1}^t f_s^u w^{u-t}$, where $f_s^t = 1$ if the strategy s is the minimum strategy in game group j at period t , otherwise $f_s^t = 0$.

[Input 6, π_{ij}^T] the weighted sum of obtained payoffs in the last T periods: $\pi_{ij}^T = \sum_{u=t-T+1}^t \pi_{ij}^u w^{u-t}$, where π_{ij}^u is the payoff of agent ij at period u .

In our artificial genetic system for simulations, the minimum strategy coordination game is played by n agents in each of m game groups. Therefore, there are m agents for

each type of players. There are \bar{s} alternative strategies, and each of the agents chooses one among them. Then agents obtain payoffs defined by the payoff function (II). Repeatedly playing the game, agents obtaining larger payoffs are likely to survive; if this is not the case, such agents are easily culled out.

We start by describing how the parameters prescribing an agent are initialized. In the experiment conducted by [Van Huyck et al. \(1990\)](#), subjects understand the payoff table defined by the payoff function (II), and it is not true that they start to play the game without any prior knowledge of the game. Therefore it is natural for artificial agents in our system to have some knowledge of the game before playing it. To do so, by using the error back propagation algorithm with the teacher signals which are shown below, we adjust the parameters of the synaptic weights and the thresholds in the neural network.

After initializing the parameters of the neural networks, the artificial agents repeatedly play the game, and accumulate the payoffs obtained in the iterated games.

The whole population is divided into m groups for playing the game. After playing the game, a subpopulation is formed by picking one agent from each group for the game; the i th subpopulation consisting of m agents corresponds to player i . The genetic operations are executed to each subpopulation separately. A fitness of each agent is a normalized payoff obtained by itself

$$f(\pi_{ij}) = \frac{\pi_{ij} - \min}{\max - \min}, \quad (2)$$

where max and min are, respectively, the maximum and the minimum payoffs in the subpopulation.

The agent choosing the strategy with the largest expected payoff is selected out of the set of candidates as the elite to be certainly reproduced in the next generation. As a reproduction operator, the roulette wheel selection is adopted. A single-point crossover operator is applied to any pair of chromosomes with the probability of crossover. With a given small probability of mutation, each gene which represents a synaptic weight or a threshold in a chromosome is randomly changed.

4 Results of the Simulations

First, we provide the result of the simulation for examining sensitivity of the parameters of the payoff function. Second, we investigate the effect of information given to players. Finally, we examine the influence of the number of players on behavior of players. The following simulations are arranged.

Simulation Coefficients: The coefficients of the payoff function.

Simulation Information: Information about outcomes of the game given to players.

Simulation Size: The number of agents playing the game.

From the space limitation, we show only the result of simulation *Coefficients* in which one treatment is described in detail but the other two treatment are summarized.

Simulation *Coefficients* is composed of three treatments: treatment *b* is carried out by varying the value of *b*, fixing the value of *a*; in treatment *a*, the value of *a* is varied, fixing the value of *b*; finally in treatment *experience*, after putting agents in experiencing the payoff dominant equilibrium, fixing the value of *a*, a treatment similar to treatment *b* is performed.

4.1 Treatment *b*

In general as the value of *b* is made larger and the risk of paying the penalty increases, the payoff of an agent selecting a large strategy such as the payoff dominant strategy 7 becomes a small value, and therefore it is likely to fail in coordination. However, the risk-free game with $b = 0$ is not the case. In this treatment, fixing the value of *a* at $a = 0.2$, the value of the penalty coefficient *b* is varied; it is set at $b = 0.0, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1$. From the data observed in the treatment, we investigate transitions and steady states of the choice rate of each strategy, the realization rate of each individual equilibrium, and so forth.

Figures 1, 2, 3 and 4 show the choice rate of each strategy, the minimum strategy rate of each strategy, the means of selected strategies and the minimum strategies, and the normalized average payoff, respectively. For comparison, in Figures 1 and 2 the data from the experiment with human subjects by Van Huyck *et al.* (1990) are provided by outline symbols. The game is played 10 times in the experiment; the data in the figures are the outcomes at the final 10th round. Moreover, the realization rate of each individual equilibrium and the gross realization rate of equilibria are given in Figures 5 and 6, respectively. The realization rates of the equilibria $(5, \dots, 5)$ and $(6, \dots, 6)$ are very small and therefore they are omitted in the figure.

In Figure 1, the choice rate of each strategy at the steady state is given. When the penalty is relatively large, i.e., $b \geq 0.04$, the secure strategy 1 is likely to be selected. Namely, most of the artificial agents avoid the risk of paying a large penalty and select the most secure strategy. As the value of *b* decreases and therefore the penalty becomes small, the modal strategy, which is the most frequently selected strategy, grows large from the strategy 1 to the strategy 4 one by one. When *b* is smaller than around 0.006, the modal strategy jumps straight to the strategy 7, and strategies 5 and 6 do not become modal.

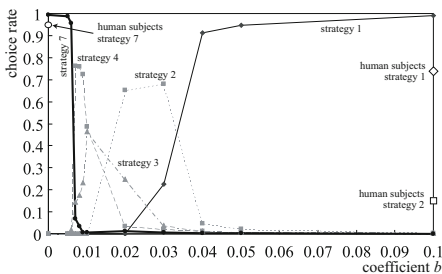


Fig. 1. Choice rate of each strategy in treatment *b*

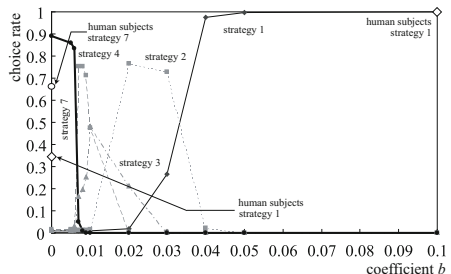


Fig. 2. Minimal strategy rate of each strategy in treatment *b*

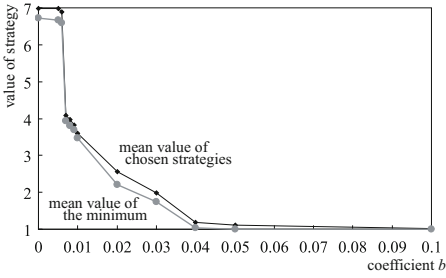


Fig. 3. Means of selected strategies and the minimal strategies in treatment b

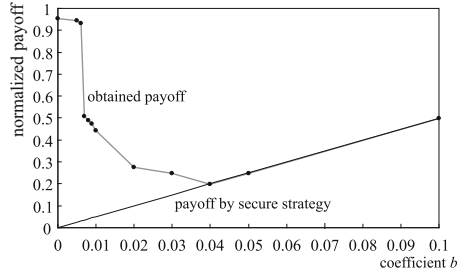


Fig. 4. Normalized average payoff in treatment b

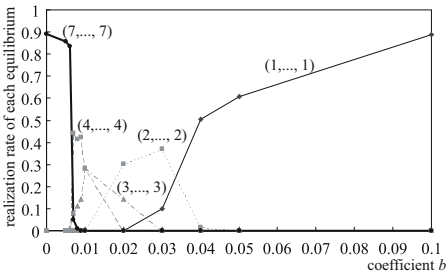


Fig. 5. Realization rate of each individual equilibrium in treatment b

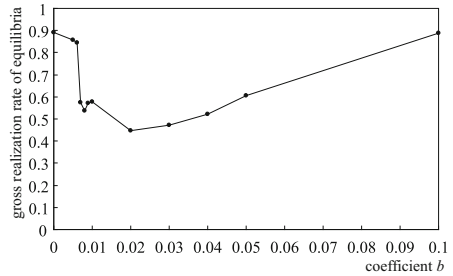


Fig. 6. Gross realization rate of equilibria in treatment b

In Figure 2, the minimum strategy rate of the strategy s means the rate that the strategy s is the minimum in the game. From the fact that Figure 2 is highly similar to Figure 1, it follows that the modal strategy in the steady state is almost the same as the minimum strategy. For the strategy 1, when $b \geq 0.04$, although the choice rate of the strategy 1 shown in Figure 1 decreases little by little as the value of b becomes small, the minimum strategy rate of the strategy 1 shown in Figure 2 is almost 1.0. Contrary to the strategy 1, when $b = 0$, the choice rate of the strategy 7 is almost 1.0, but the minimum strategy rate falls below 0.9 because the other strategies are selected on rare occasions.

Compared with the result of the experiment with human subjects, when $b = 0.1$, the choice rate of the secure strategy 1 of the artificial agents, 0.99, is larger than that of the human subjects, 0.72; for the minimum strategy rate, both of them get the highest rate, 1.0. For the case of $b = 0$, in the experiment with human subjects, the value of the gain coefficient a is set at $a = 0.1$, which is slightly different from the setting of the simulation. The choice rate of the payoff dominant strategy 7 of the artificial agents is 0.99 which is close to the result of the human subjects, 0.956; the minimum strategy rate of the artificial agents, 0.891, is larger than that of the human subjects, 0.667. All in all, the result of the simulation is similar to that of the experiment with human subjects, and therefore the result of the simulation supports that of the experiment with

human subjects. To be more precise, in the both results, the secure strategy is dominant when the risk of paying a large penalty is high; in the absence of such risk, the payoff dominant strategy is likely to be chosen. From the other perspective on this similarity, the simulation system successfully emulates the human behavior in the game.

In Figure 3 the means of the chosen strategies and the minimal strategies are shown; it can be found that these values are very similar. This fact means that at the steady state, most of the artificial agents choose the minimal strategies. From Figure 4 the payoff obtained by an agent decreases as the value of b increases from $b = 0$. At the point of $b = 0.04$, the payoff is equal to the payoff of the secure strategy 1. Because when $b \geq 0.04$, the payoff of the secure strategy 1 grows large with the value of b , we can understand that most of the artificial agents choose strategy 1 in such a situation.

The realization rate of each individual equilibrium is given in Figure 5. When $b = 0.1$, the secure equilibrium $(1, \dots, 1)$ is realized at the rate of 0.89 in the steady state. Although as the value of b decreases, the realization rate of the secure equilibrium decreases, it should be noted that in the interval $0.04 \leq b \leq 0.1$, only the secure equilibrium $(1, \dots, 1)$ is realized. As the value of b still decreases over 0.04, the consecutive equilibria, $(2, \dots, 2)$, $(3, \dots, 3)$, and $(4, \dots, 4)$, can be found, but the realization rates of these equilibria do not exceed 0.5. When $b \leq 0.006$, the payoff dominant equilibrium $(7, \dots, 7)$ is realized at the rate larger than 0.8.

The gross realization rate of equilibria is shown in Figure 6; it is found that at both ends of the horizontal axis, $b = 0$ and $b = 0.1$, the equilibria are likely to be realized. In the intermediate cases where effectiveness of the risk of paying the penalty is not clear, it becomes difficult for artificial agents to coordinate their strategies, and therefore the gross realization rate of equilibria descends, compared with the cases of $b = 0$ and $b = 0.1$.

The result of treatment b can be summarized as follows.

1. In the games without the risk of paying any penalty, the artificial agents successfully coordinate their strategies and the payoff dominant equilibrium is realized.
2. In the games with the risk of paying a substantial penalty, coordination among the artificial agents is failed, but they suitably predict strategies of the opponents and the secure equilibrium forms.
3. The games with the risk of paying the intermediate penalty are likely to bring outcomes of disequilibria.
4. As the value of b decreases, artificial agents shift choices of strategies stepwise from the secure strategy 1 to the payoff dominant strategy 7.
5. While the payoff dominant equilibrium is sensitive to increase of the value of b , the secure equilibrium is not so sensitive to decrease of the value of b .

4.2 Treatment a

While in treatment b the value of the penalty coefficient b is varied fixing the value of a at 0.2, in this treatment conversely fixing the value of b at 0.0 and 0.1, the value of the gain coefficient a is changed; the value of a is set at $a = 0.1, 0.2, 0.5, 1.0, 1.5, 2.0$. Especially, for the case of $b = 0.1$, we additionally vary the value of a from $a = 2.0$ to $a = 5.0$ at intervals of 0.5. We examine the data of the treatment as in treatment b .

The result of treatment a can be summarized as follows.

1. When $b = 0$, the artificial agents successfully coordinate their strategies, and as a result the payoff dominant equilibrium is realized.
2. In the case of $b = 0.1$, as the value of a increases, most of the artificial agents shift choices of strategies from the secure strategy 1 to the payoff dominant strategy 7 by way of the strategies 3 and 4.
3. When $a \geq 4.0$ where the effect of the risk of paying the penalty is relatively small, the artificial agents successfully coordinate their strategies and the payoff dominant equilibrium occurs at the rate of about 0.9.
4. When $a = 0.2$ where the effect of the risk of paying the penalty is relatively large, coordination of the artificial agents is failed but they successfully predict opponents' strategies; consequently the secure equilibrium is realized at the rate of about 0.9.
5. The games with $0.5 \leq a \leq 3$ where the risk of paying the penalty is intermediate are likely to bring outcomes of disequilibria.
6. The behavior of the artificial agents roughly depends on the ratio of the penalty coefficient b to the gain coefficient a ; however, around $a/b = 10$, the influence of the penalty coefficient b seems to be larger.

4.3 Treatment Experience

In the experiment by [Van Huyck et al. \(1990\)](#), by playing the game with $b = 0$ where there does not exist any risk of paying the penalty, the subjects experience the payoff dominant equilibrium. Subsequently without interruption they also play the game with $b = 0.1$ where there exists the risk of paying the substantial penalty. By doing this, they examine whether or not the experience of the payoff dominant equilibrium can avoid the coordination failure. The result is negative; the subjects cannot keep the coordination, and eventually their strategies converge at the secure strategy. To be more precise, the choice rate of the payoff dominant strategy is 0.96 at the end of the $b = 0$ session, but conversely the choice rate of the secure strategy becomes 0.84 at the end of the $b = 0.1$ session.

In our simulation, fixing the value of a at 0.2, after the risk-free game with $b = 0$ is played, the risky game with $b = 0.1, 0.05, 0.01$, or 0.005 follows. Through this treatment, we examine influence of the experience of the payoff dominant equilibrium on behavior of artificial agents. We provide two cases: one includes the learning process to impress the artificial agents on modification of the payoff function, and the other does not include it. The learning process is implemented by using the error back propagation algorithm in a way similar to the learning of the initial population.

The result of treatment *experience* can be summarized as follows.

1. When $b = 0.1$, the experience of the payoff dominant equilibrium does not influence behavior of the artificial agents as in the experiment with human subjects.
2. When $b = 0.05$, $b = 0.01$, and $b = 0.005$, the experience of the payoff dominant equilibrium prompts the artificial agents to select larger strategies.

3. For cases without the learning process, although it requires a long time compared with the case with the learning process when $b = 0.1$, the secure strategy 1 becomes dominant in the population; when $b = 0.05$, $b = 0.01$, and $b = 0.005$, most of the artificial agents continue to choose the payoff dominant strategy 7.

5 Conclusions

We have focused on the coordination games characterized by the minimum strategy used in the experimental investigation by [Van Huyck et al. \(1990\)](#), and developed a simulation system, where artificial adaptive agents have a mechanism of decision making and learning based on neural networks and genetic algorithms, for the extensive range of analysis on behavior of artificial agents in the coordination games in order to complement the experiments with human subjects.

We summarize the findings from the simulation analysis as follows. (i) In treatment b of Simulation *Coefficients*, the result of the simulation supports that of the experiment with human subjects on the whole. As new insights from this simulation analysis, it is found that while the payoff dominant equilibrium is sensitive to increase of the value of the penalty coefficient b , the secure equilibrium is not so sensitive to decrease of the value of b . Moreover, in the games without the penalty or the games with a substantial penalty, the equilibria are likely to be realized; the games with the risk of paying an intermediate penalty are likely to bring outcomes of disequilibria. (ii) From treatment a of Simulation *Coefficients*, the behavior of the artificial agents roughly depends on the ratio of the penalty coefficient b to the gain coefficient a . (iii) If the penalty coefficient b is not so large, the experience of the payoff dominant equilibrium prompts the artificial agents to select larger strategies. (iv) The data from Simulation *Information* does not confirm the finding of the experiment with human subjects that disclosure of the distribution of strategies accelerates the convergence to selecting the secure strategy 1, but it is found that there exist some different effects of the additional information on the behavior of agents. (v) In Simulation *Size*, we find out that the equilibria are likely to be realized as the number of players decreases. The fact from the simulation supports the result of the experiment with human subjects.

Reference

- Battalio, R., Samuelson, L., Van Huyck, J.: Optimization incentives and coordination failure in laboratory stag hunt games. *Econometrica* 69, 749–764 (2001)
- Cooper, R.W., DeJoong, D.V., Forsythe, R., Ross, T.W.: Selection criteria in coordination games: some experimental results. *American Economic Review* 80, 218–233 (1990)
- Friedman, D.: Equilibrium in evolutionary games: some experimental results. *The Economic Journal* 106, 1–25 (1996)
- Goeree, J.K., Holt, C.A.: An experimental study of costly coordination. *Games and Economic Behavior* 51, 349–364 (2005)
- Morgan, J., Sefton, M.: An experimental investigation of unprofitable games. *Games and Economic Behavior* 40, 123–146 (2002)
- Ochs, J.: The coordination problem in decentralized markets: an experiment. *The Quarterly Journal of Economics* 105, 545–559 (1990)

- Schmidt, D., Shupp, R., Walker, J.M., Ostrom, E.: Playing safe in coordination games: the role of risk dominance, payoff dominance, and history of play. *Games and Economic Behavior* 42, 281–299 (2003)
- Straub, P.G.: Risk dominance and coordination failures in static games. *The Quarterly Review of Economics and Finance* 35, 339–363 (1995)
- Van Huyck, J.B., Battalio, R.C., Beil, R.O.: Tacit coordination games, strategic uncertainty, and coordination failure. *American Economic Review* 80, 234–248 (1990)
- Van Huyck, J.B., Battalio, R.C., Beil, R.O.: Strategic uncertainty, equilibrium selection, and coordination failure in average opinion games. *The Quarterly Journal of Economics* 106, 885–910 (1991)

A Genetic Algorithm for Community Formation Based on Collective Intelligence Capacity

Emil Scarlat and Iulia Maries

University of Economics, Economic Cybernetics Department, 15-17 Dorobanti Avenue,
Sector 1, Bucharest, Romania
emil_scarlat@yahoo.com, iulia.maries@hotmail.com

Abstract. Community formation has certainly gained more and more attention from both the researchers and practitioners in the field of complex networks. An efficient algorithm is needed since the number of the possible communities is exponential in the number of agents. Genetic algorithm is a very useful tool for obtaining high quality and optimal solutions for optimization problems, due to its self-organization, self-adaptation and parallelism. The paper proposes a high performance genetic algorithm for community formation. The key concept in our algorithm is a new fitness index, which aims at being a trade-off between intelligence and cooperation, and allows not only community formation but also intelligence to be driving principle in the community formation process.

Keywords: Genetic Algorithms, Collective Intelligence, Intelligence Index, Communities of Practice.

1 Introduction

Complex systems composed of different interacting subsystems tend to evolve towards more coherence and interdependence as the subsystems mutually adapt. The continuous evolution, adaptation, cooperation and negotiation determine an increasingly diverse, complex and efficient organization [3].

To successfully deal with organizational problems we need to develop collective intelligence as a global civilization. Collective intelligence improves competitiveness within organizations in the context of a global market and collective performance represents a critical factor in the organization's development. In this situation creating, developing and sustaining trust among the members of a community is the core that leads to performance.

The remainder of this paper is organized as follows: the concept and the emergence of community of practice are presented in Section 2. In the same framework, collective intelligence is defined in Section 3. Section 4 presents genetic algorithms in the context of cluster and partition analysis. In Section 5 we propose a genetic algorithm for community formation based on collective intelligence capacity. Finally, Section 6 comes up with conclusions, relevant implications and directions for future work.

2 The Emergence of Communities of Practice

A new organizational form has emerged that complements existing structures and emphasizes knowledge sharing and learning, called the community of practice. The

concept has received much attention from researchers and practitioners in the management area, especially in knowledge management.

The concept “community of practice” was outlined by Lave and Wenger in early 90’s, to describe “a group of people who share a concern, a set of problems or a passion about a topic, and who deepen their knowledge and expertise by interacting on an ongoing basis” [18]. In this context, communities of practice are free associations of persons who choose to improve their skills together. Furthermore, they develop new productive collective capabilities, which are sources of value creation in knowledge-based economies. Communities of practice are groups of self-governing people who are challenged to create value by generating knowledge and increasing capabilities (Fig. 1).

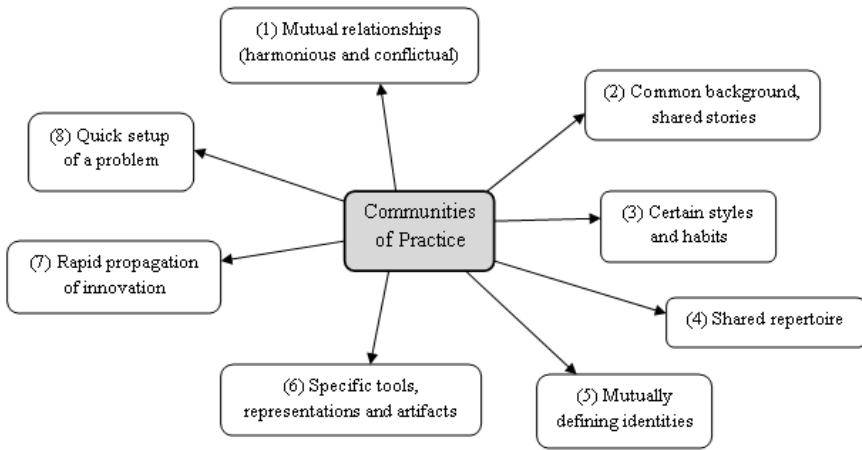


Fig. 1. The key characteristics of communities of practice are presented (Source: compiled from [18])

Inside communities of practice the individual knowledge and experiences are shared, new knowledge is developed and problems are solved through interactions between members of the communities.

A community of practice represents “an intrinsic condition for the existence of knowledge” [7]. In general, knowledge sharing is taking place in non-hierarchical groups, beyond the formal work description. This phenomenon has been identified as the “emergent communities of practice”. The emergent communities of practice are created by the complexity, diversity and fluidity of the individuals.

The core participants of the community develop common cognitive frames and knowledge. This facilitates coordination among community members, but also restricts the capacity to find solutions to new problems. Original solutions to problems are given by the peripheral members. The process of learning at the boundary is the base of the legitimacy of peripheral members [16]. The evaluation of relevant knowledge and information, and also of “best practices” related to the practice represents

the most important activities in communities of practice. Through these activities, members are able to engage and participate in collective learning process. Therefore, many communities of practice emerge naturally from existing relationships and trustfulness.

It seems that the emergence of centralized structures in the communities of practice is the outcome of a self-organizing process [11]. In this context, temporary organizations are motivated to perform specific actions in order to achieve immediate goals. The emergence of such temporary and network-based organizations has determined the creation of communities of practice, which become active temporary in network-based organizations.

In such a dynamic environment, these characteristics represent conditions for the input and output of the community process. Therefore, a successful community depends on the degree to which the processes mutually match each other.

3 Collective Intelligence

Collective intelligence is a shared intelligence that emerges from the collaboration of individuals. The researchers in collective intelligence field consider it as a subfield of sociology, communication or behavior, computer science or cybernetics.

3.1 Intelligence

Intelligence is related to the goal or problem and to the previous knowledge and experience of the solver. In our context, intelligence is defined as “the ability for attaining goals of for solving problems that puts at work responsiveness of the solver to the situation where the goal or problem arises and use of its previous knowledge and experience.” [2]

3.2 Intelligence of Collectivities

“Collective intelligence is neither a new concept nor a discovery. It is what shapes social organizations – groups, tribes, companies, teams, governments, nations, societies, guilds, etc... - where individuals gather together to share and collaborate, and find an individual and collective advantage that is higher than if each participant had remained alone. Collective intelligence is what we term a positive-sum economy.” [12] So the presence of collective intelligence has been felt for a long time: families, companies and states are groups of individuals that at least sometimes act intelligent. There are groups of insects, such as bee and ant colonies, that are finding food sources acting intelligent. Also, the human brain could be seen as a collection of individual neurons that collectively act intelligent.

Collective intelligence can be defined as a group ability to solve more problems than its individuals [6]. In order to overcome the individual cognitive limits and the difficulties of coordination, a collective mental map can be used. A collective mental map is represented as an external memory with shared access that can be formalized as a directed graph. The efficiency of mental problem-solving depends on the problem

representation in the cognitive system. Intelligent agents are characterized by the quality of their mental maps, knowledge and understanding of their environment, capacities for action or goals.

Collective intelligence offers a new perspective to different phenomena. This concept suggests another way of thinking about effectiveness, profitability, teamwork or leadership. The formal hierarchies of traditional organizations need to be replaced by self-organizing communities of practice. Why? Because “most fields of expertise are now too complex for any one person to master and thus collective intelligence must be brought to bear to solve important problems”.

4 Genetic Algorithms for Cluster and Partition Analysis

A genetic algorithm represents an iterative process that applies genetic operators such as selection, crossover and mutation to a population of elements. The elements, called chromosomes, represent possible solutions to the problem.

Each chromosome has associated a fitness value which quantifies its value as a possible solution. Obviously, a chromosome representing a better solution will have a higher fitness value. The chromosomes compete for survival based on their fitness value. The crossover operator transfers genetic material from one generation to another. The mutation operator introduces new genetic material into the population.

Genetic clustering or partitioning algorithms include selection, crossover and mutation operators, adaptations of these operators and also some totally different operators. Adaptation is essential, well defined fitness functions and suitable operators are needed to encode potential solutions and to induce the evolution towards the optimal clustering.

Chromosomes are selected for reproduction based on their fitness function. If all these fitness values are positive and the maximum fitness value corresponds to the optimal clustering, then fitness selection might be appropriate [8]. Otherwise, a ranking selection method might be used [4]. In this context, mutation corresponds to moving an object from one cluster to another.

When the number of clusters is unknown, the problem becomes a general clustering problem and the number of clusters should be determined. The most important parameter, the number of clusters, becomes an upper limit (C_{max}) of the current number of clusters.

Usually, genetic algorithms are used to identify the cluster centroids. The chromosomes encode the coordinates of the centers of k desired clusters. The method allows both binary and floating point representations, clustering being performed either in crisp or fuzzy mode.

Another application of genetic algorithms on clustering problems proposes different encoding of solutions in chromosomes. Each position in the chromosome represents an item in the training set. This partition-based encoding method determines successive clustering generations to find the right cluster for each data item.

The community formation problem is a partitioning problem, aiming to find good partitions of a set of agents into disjoint communities. The solution of a problem must satisfy various constraints, otherwise the solution is invalid. The objective of the grouping is to optimize the fitness function.

Classical genetic algorithms cannot be directly applied to partitioning problems. The structure of the chromosomes is item oriented instead of group (community) oriented, so a special encoding scheme is needed to transfer the relevant structures of grouping problems into genes in chromosomes.

5 A Community Formation Mechanism Using Genetic Algorithms

The encoding scheme focuses on transferring the genes into relevant groups or communities. This encoding scheme ensures both the transmission of the genes from one generation to the next and a better quality estimation of the regions they occupy in the search space.

In this context, a chromosome can be represented as a set of a number of mutually disjoint communities:

$$\{a_{i_1}, \dots, a_{i_{k(1)}}\} \dots \{a_{i_{n-k(C)+1}}, \dots, a_{i_n}\} \tag{1}$$

where $k(j), j = 1, \dots, C$, denotes the length of the community j (the number of agents in that community) and C the number of communities encoded in a chromosome.

5.1 The Fitness Index

The most important part is to find a measurement of the suitability of an agent i into the community c based on the execution of task j . We called this measurement *intelligence score* (Table 1). The intelligence score is denoted by μ_{ij} , where $0 \leq \mu_{ij} \leq 100$.

The intelligence scores measure agents' intelligence in executing the current tasks by forming communities, combining agents with the tasks based on their intelligence.

The overall performance of a community c in respect to a task j is defined by an *intelligence index*, denoted μ_j^c .

Communities for which the intelligence of completing the task j matches exactly or exceeds the necessary capacity are valued to 1. The ones for which the intelligence does not match the necessary capacity are valued to 0. The *intelligence index* has the following form:

$$\begin{cases} 1, & \text{if } \sum_{i \in c} \mu_{ij} \geq 100 \\ 0, & \text{if } \sum_{i \in c} \mu_{ij} < 100 \end{cases} \tag{2}$$

The aggregate *intelligence index* of the community c for completing the overall task is calculated as:

$$\mu^c = \prod_{j=1}^p \mu_j^c, 0 \leq \mu^c \leq 1 \tag{3}$$

Table 1. The intelligence scores of n agents participating in p tasks

	<i>Task 1</i>	...	<i>Task j</i>	...	<i>Task p</i>
<i>Agent 1</i>	μ_{11}	...	μ_{1j}	...	μ_{1p}
...
<i>Agent i</i>	μ_{i1}	...	μ_{ij}	...	μ_{ip}
...
<i>Agent n</i>	μ_{n1}	...	μ_{nj}	...	μ_{np}

The *intelligence index* of the partitioning solution (partition of n agents into C communities) represents the average μ^c values:

$$f = \frac{\sum_{c=1}^C \mu^c}{c} = \frac{\sum_{c=1}^C \prod_{j=1}^p \mu_j^c}{c}, 0 \leq f \leq 1 \tag{4}$$

where C represents the number of communities in the solution.

In this context, the problem of community formation can be formulated as follows:

$$\max_C \frac{\sum_{c=1}^C \mu^c}{c} \tag{5}$$

Our approach proposes a new fitness index, which aims at being a trade-off between intelligence and cooperation. This “intelligence index” allows community formation, partitioning n agents into C communities, and also collective intelligence, emerging from the collaboration of agents, to be driving principle in the community formation process.

5.2 Numerical Example

We have run our experiments on synthetic data. The experiments have shown up that genetic algorithms provide the capability of producing optimal partitions of a set of agents.

To illustrate the measurements and indexes mentioned above we will present a numerical example.

Table 2. The intelligence scores of 5 agents participating in 3 tasks

	<i>Task 1</i>	<i>Task 2</i>	<i>Task 3</i>
<i>Agent 1</i>	60	44	83
<i>Agent 2</i>	62	22	79
<i>Agent 3</i>	37	89	75
<i>Agent 4</i>	84	90	67
<i>Agent 5</i>	79	48	70

Table 3. The intelligence scores of 2 communities with respect to 3 tasks

	<i>Task 1</i>	<i>Task 2</i>	<i>Task 3</i>
<i>{1,4,5}</i>	$60+84+79 = 223$	$44+90+48=182$	$83+67+70=220$
<i>{2,3}</i>	$62+37=99$	$22+89=111$	$79+75=154$

Obviously, the first community has a better performance than the second one. The intelligence score of each task in the first community are greater than the ones in the second community.

$$\mu_1^1 = 1, \mu_2^1 = 1, \mu_3^1 = 1 \rightarrow \mu^1 = 1 \cdot 1 \cdot 1 = 1$$

$$\mu_1^2 = 0, \mu_2^2 = 1, \mu_3^2 = 1 \rightarrow \mu^2 = 0 \cdot 1 \cdot 1 = 0$$

$$f = (1 + 0) / 2 = 0.5$$

The numerical example seems to be enlightening. Even so, collective intelligence should not be perceived as the sum of individual intelligences, it represent the ability of the community to complete more tasks than its single individuals.

6 Conclusions and Future Work

Communities of practice are the core of collective learning and collective intelligence, relaying on a permanent exchange of knowledge and information related to the practice. They enhance particular knowledge and contribute to its coherence. Communities of practice can provide a social reservoir for practitioners and knowledge producers to analyze, address and explore new solutions to their problems.

In the context of the knowledge-based society, individual intelligence cannot face all the problems. To successfully deal with problems we need to develop collective intelligence as a global civilization. Thus developing and sustaining collaboration among team members is the core that leads to performance.

Our research is based on the theoretical approaches presented in the literature, with emphasis on genetic algorithms applications. The paper proposes a genetic algorithm for community formation based on collective intelligence capacity. This approach introduces the concept of intelligence index, aiming to optimal partitions of a set of agents. The mechanism highlights the relevance of intelligence in community formation and reveals the need for such mechanisms that allow large group of professionals to make decisions better than single individuals. Our future work in this direction will focus on extending the experiments of genetic algorithms in community formation and applying the algorithm on real data.

Acknowledgement. This article is one of the results of the research activity carried out by the authors under the frame of following projects:

1. PNII – IDEI project no. 810/2008 funded by CNCISIS – UEFISCDI – “The Development of Theoretical Fundamentals of Knowledge Ecosystems and Their Applications on Economy and Healthcare”, coordinated by Professor Virginia Maracine;
2. “Doctoral Program and Ph.D Students in the education, research and innovation triangle”. This project is co-funded by European Social Fund through the Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by the Academy of Economic Studies, Bucharest.

References

1. Andriessen, E., Huis, M.: Group dynamics and CoPs. In: Position paper for ECSCW 2001 Workshop 6, Actions and Identities in Virtual Communities of Practice (2001)
2. Garrido, P.: Business Sustainability and Collective Intelligence. *The Learning Organization* 16(3), 208–222 (2009)
3. Georgescu, V.: Evolving Coalitions of Task-Oriented Agents via Genetic Algorithms to Foster Self-Organization in Digital Business Ecosystems. In: Proceedings of the International Conference on Modeling Decision for Artificial Intelligence (2007)
4. Golberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, Reading (1989)
5. Gruszczyk, W., Kwasnicka, H.: Coalition Formation in Multi-Agent Systems – An Evolutionary Approach. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 125–130 (2008)
6. Heylighen, F.: *Collective Intelligence and its Implementation on the Web: algorithms to develop a collective mental map*. Springer, Netherlands (1999)

7. Juriado, R., Gustafsson, N.: Emergent communities of practice in temporary inter-organizational partnerships. *The Learning Organization: The International Journal of Knowledge and Organizational Learning Management* 14(1), 50–61 (2007)
8. Krovi, R.: Genetic Algorithms for Clustering: A Preliminary Investigation. In: *Proceedings of the 25th International Conference on System Sciences*, pp. 540–544 (1992)
9. Lave, J., Wenger, E.: *Situated Learning. Legitimate Peripheral Participation*. Cambridge University Press, Cambridge (1991)
10. Luck, M., McBurney, P., Preist, C.: *Agent Technology: Enabling Next Generation Computing – A Roadmap for Agent Based Computing* (2003); ISBN 0854 327886
11. Muller, P.: Reputation, trust and the dynamics of leadership in communities of practice. *J. Manage Governance* (2006)
12. Noubel, J., F.: Collective intelligence, the invisible revolution. *The Transitioner.org* (2004),
http://www.thetransitioner.org/wen/tiki-list_file_gallery.php?galleryId=1
13. Por, G., van Bukkum, E.: *Liberating the Innovation Value of Communities of Practice*, Amsterdam (2004)
14. Scarlet, E., Maries, I.: Towards an Increase of Collective Intelligence within Organizations Using Trust and Reputation Models. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS*, vol. 5796, pp. 140–151. Springer, Heidelberg (2009); ISSN 0302-9743
15. Szuba, T., Almulla, M.: Was Collective Intelligence before Life on Earth? In: *IPDPS Workshops on Parallel and Distributed Processing* (2000)
16. Turoff, M., Hiltz, S.R.: The future of professional communities of practice. In: Weinhardt, C., Luckner, S., Stößer, J. (eds.) *WEB 2008. LNBIP*, vol. 22, pp. 144–158. Springer, Heidelberg (2009)
17. Wenger, E.C., Snyder, W.M.: Communities of practice: the organizational frontier. *Harvard Business Review* 78(1), 139–145 (2000)
18. Wenger, E., McDermott, R.A., Snyder, W.: *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Harvard Business School Press, Boston (2002)
19. Zhang, W., Watts, S.: Online communities as communities of practice: a case study. *Journal of Knowledge Management* 12(4), 55–71 (2008)

Argumentative Agents Negotiating on Potential Attacks

Guido Boella¹, Dov M. Gabbay², Alan Perotti¹,
Leendert van der Torre³, and Serena Villata¹

¹ Dipartimento di Informatica, University of Turin, Italy
{guido,perotti,villata}@di.unito.it

² King's College London, UK
dov.gabbay@kcl.ac.uk

³ CSC, University of Luxembourg
leendert@vandertorre.com

Abstract. When arguing, agents may want to discuss about the details after agreeing about the general problems. We propose to model this kind of situation using an extended argumentation framework with potential attacks. Agents negotiate about raising potential attacks or not, in order to maximize the number of their accepted arguments. The result of the negotiation process consists in the formation of coalitions composed by those agents which have found an agreement. The two proposed negotiation protocols have been implemented and an evaluation, addressed by means of experimental results, shows which combination of strategies and negotiation protocol allows the agents to optimize outcomes.

1 Introduction

Agents communicate with each other by exchanging the arguments on which their beliefs are grounded. The degree of information each argument brings to the agents can be subject of debate. Consider the well known *blocks world* environment. Communicating robots discuss about their plans and their joint feasibility. For instance, suppose that an agent asserts her will to *put down block A* while another agent states that she will *stack block B on block C*. These two arguments address only general issues while the matter is how each party intends to “refine” them. Boella *et al.* [2] propose to model the refinement process by extending an abstract argumentation framework with potential, or voluntary, attack relations. The aim of this paper is to analyze how these attacks impact on a multiagent system, where agents have to consider the consequences of their actions in the social context.

We address the following research questions: *Which negotiation protocols are suitable? Which strategic behaviors can agents adopt? What are coalitions in this context?* We briefly resume the concept of extended argumentation framework with voluntary attacks, then we present two negotiation protocols for deciding whether to carry out potential attacks and several strategic behaviors. Finally, we define coalitions as a side-effect of the negotiation process, i.e. as a structure

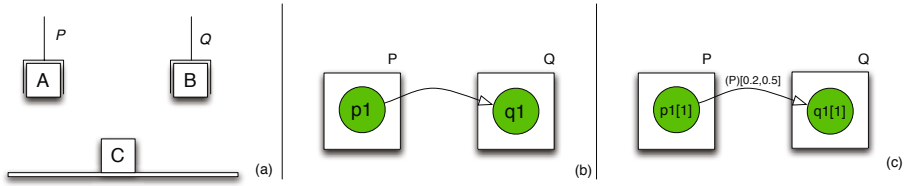


Fig. 1. Robots in the blocks world and their arguments

which is created when a solution is achieved during the negotiation. Experiments and conclusions end the paper.

2 Agents and Arguments

Given a Dung argumentation framework [4] $AF = \langle A, R \rangle$, an extended argumentation framework with potential attacks AF^+ is a triple $\langle A, R, P \rangle$, with a set of abstract elements called *arguments* A , and two disjoint binary relations, called *attack* and *potential attack*, $R, P \subseteq A \times A$, $R \cap P = \emptyset$. Intuitively, a potential attack indicates the possibility that, in the evolution of an argumentation framework, an attack relation is established among two arguments. A potential attack is translated into a classical attack when it is concretized, or it is removed from the framework when it is chosen not to insert it.

We assume to have a distributed and decentralized set of agents AG . Each agent controls a set of arguments of the AF^+ as expressed by the relation $Ctrl_{arg} \subseteq A \times AG$. Arguments represent the objectives of the agents, the proposals they address to the other agents. We define also the relation $Ctrl_{att} \subseteq P \times AG$, which associates every potential attack to the agents which can activate or deactivate it. The goal of the multiagent system is to gradually obtain a grounded framework where all potential attacks have been either concretized or removed.

Example 1. Consider the example visualized in Figure 1a: two robots, P and Q , operate in the blocks world. P holds block A and Q holds block B , while block C lies on the table. Suppose that P 's goal is to *put down block A* and Q 's is to *put block B on block C*. P can achieve her goal by putting block A on the table or on block C , and therefore her argument can be refined to *put block A on block C* or *put block A on the table*. Now suppose that Q cannot unstack A from C before putting B down on C due to some resource limitation (the set of actions may require too much time or battery power). Therefore one of P argument's refinements attacks Q 's argument and P 's argument potentially attacks Q 's argument. P controls the $(p1, q1)$ attack. Figure 1b is the extended argumentation framework.

Given an AF^+ , every possible framework achievable from a potential attack activation/deactivation is a *scenario*. A *final scenario* is a scenario which does not contain potential attacks. Each agent ag_i evaluates the desirability of a

scenario taking into account the acceptability of its arguments and the values of parameters α_i and β_i .

Definition 1. (Agents’ parameters)

- Parameter α_i^j is associated with each argument j of agent ag_i and it describes the evaluation of j from ag_i .
- Parameters $\beta_{i+}^{j,k}$ and $\beta_{i-}^{j,k}$ are defined if $j, k \in A$ and $(j, k) \in P$ and $Ctrl_{att}((i, j), ag_i)$.
- Parameter γ_i is the relative sociality of ag_i .

Parameter α_i^j allows the agents to have preferences on the arguments they own. Notice that this preference relation is defined for the couple agent-argument and it does not depend on the AF^+ , i.e., on the acceptability of the arguments. Parameters $\beta_{i+}^{j,k}$ and $\beta_{i-}^{j,k}$ model the cost associated to the activation β_{i+} or deactivation β_{i-} of the potential attack. Parameter γ_i indicates the relative sociality of the agent regarding coalitions. An agent with a high γ_i will evaluate as fundamental being part of a coalition while an agent with a low γ_i will evaluate this situation as uninfluential. We assume each agent has a complete knowledge about the AF^+ and the refinements of each argument but parameters α, β, γ are internal to the agent and they are not known by the other agents. Parameters α_i and β_i concur in the evaluation of a scenario by means of the utility functions of Definition 2.

Definition 2. (Utility functions: cost-benefit analysis)

- $ut_i^{arg}(AF_1^+) = \sum_j \alpha_i^j, \forall j \in A, L(j) = IN, ag_i \in Ctrl_{arg}(j),$
- $ut_i^{att}(AF_1^+) = \sum_{j,k} \beta_{i+}^{j,k} + \sum_{q,w} \beta_{i-}^{q,w} \forall (i, j) \in R' \setminus R \wedge ag_i \in Ctrl_{arg}(j, k), (q, w) \in (R \cup P) \setminus (R' \cup P') \wedge ag_i \in Ctrl_{arg}(q, w),$
- $ut_i^{fw}(AF_1^+) = ut_i^{arg}(AF_1^+) - ut_i^{att}(AF_1^+).$

The function ut_i^{arg} is the sum of each argument j of agent ag_i accepted in the framework. If the agent owns only rejected arguments $ut_i^{arg}(AF^+) = 0$. The function ut_i^{att} is composed by two sums, $\sum_{j,k} \beta_{i+}^{j,k}$ and $\sum_{q,w} \beta_{i-}^{q,w}$, where the first one considers the potential attacks (j, k) of ag_i activated in AF_1^+ and the second one considers the potential attacks not activated. Utility function ut_i^{fw} composes the results of the two previous utility functions. From now on, we will use the term *outcome* to indicate the utility associated by an agent to a scenario.

Example 2 (Continued). In Figure 1c, the numbers associated to the arguments represent the value of the α parameter, the tuples of numbers on the attack relations are the values of the β_-, β_+ parameters. In this example, both agents own a single argument; agent P controls the attack $(p1, q1)$, the activation cost of this attack is 0.5 while the deactivation cost is 0.2: for instance, putting block A on block C may require more battery power because it is a more specific action. Since this framework contains a single potential attack there are only two

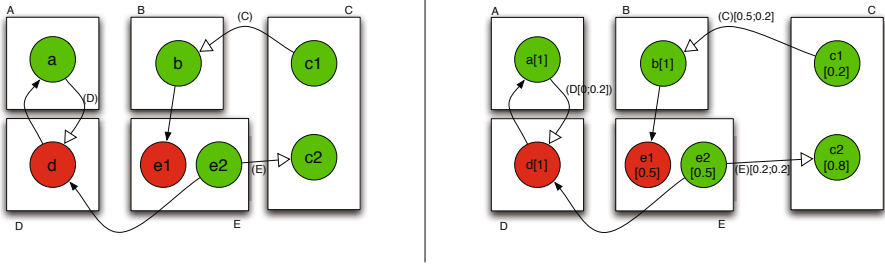


Fig. 2. AF^+ and α, β, γ parameters

possible final scenarios: in both cases agent P 's argument will be accepted (same ut_P^{arg}), but the agent will choose not to raise the potential attack because the deactivation cost is lower. Agent P could decide whether to activate her attack because there was no other agent controlling a potential attack. In more complex examples, such as the one depicted in Figure 2 agents have to take into account other agents' arguments and potential attacks, and they need to communicate and negotiate about which subset of potential attacks shall be activated. The outcomes of the agents in the framework of Figure 2 are represented in Table 1.

Definition 3. (Preferences among scenarios)

$$\forall AF_1^+, AF_2^+ \in S(AF^+), AF_1^+ \preceq_i AF_2^+ \text{ IFF } ut_i^{fw}(AF_1^+) \leq ut_i^{fw}(AF_2^+).$$

Table 1. Outcomes of the final scenarios of Figure 2

Agent	AF_0^+	AF_1^+	AF_2^+	AF_3^+	AF_4^+	AF_5^+	AF_6^+	AF_7^+
A	1	1	1	1	1	1	1	1
B	1	1	0	0	1	1	0	0
C	0.5	-0.3	0.8	0	0.5	-0.3	0.8	0
D	0	0	0	0	-0.2	-0.2	-0.2	-0.2
E	0.3	0.3	0.8	0.8	0.3	0.3	0.8	0.8

Note that different agents can have different preferences over the set of possible scenarios, so they need to interact in order to find an agreement upon which attack they shall activate or deactivate; moreover, agents' evaluation of the scenarios is not known by the other agents, so there is no way to locally compute a solution and a negotiation process has to be faced.

3 Negotiation

We introduce two negotiation protocols defined in the context of argumentation frameworks with potential attacks: GP (*global protocol*) and SP (*server-centered*

protocol). We assume the agents participate to the negotiation process only if they have the control of, at least, one potential attack. We follow the three features for automated negotiation highlighted by Jennings *et al.* [7]: negotiation objects, negotiation protocols and decisional model of the agents.

Consider the AF^+ defined in Figure 2 and the outcomes provided in Table 1. From the perspective of the agents, the relevant issue is the activation/deactivation of the potential attacks with the aim to maximize their outcome. For instance, from the point of view of agent C the relevant issue is the acceptability of argument $c2$ which means that she needs the deactivation of the potential attack $(e2, c2)$. The knowledge about which refinement of argument $e2$ of agent E causes the removal of this potential attack is irrelevant for agent C . Thus, the negotiation objects of the agents are the scenarios. This means that the negotiation proposals correspond with the scenarios. We represent the proposals as a tuple of lists of potential attacks $\langle P^+, P^- \rangle$, the potential attacks the agent desires to activate or remove, respectively. Considering the framework in Figure 2, a final scenario corresponds to the deactivation of each potential attack and is described by the negotiation proposal $\langle \{ \}, \{ (a, d), (c1, b), (e2, c2) \} \rangle$.

The negotiation objects are defined as follows:

Definition 4 (Negotiation objects). *Given $AF^+ = \langle A, R, P \rangle$, a proposal is a scenario $AF_1^+ \in S(AF^+)$. A proposal corresponding to the scenario $AF_1^+ = \langle A, R_1, P_1 \rangle$ is formalized as a tuple $\langle P^+, P^- \rangle$ such that: $P^+ = R_1 \setminus R$ and $P^- = (R \cup P) \setminus (R_1 \cup P_1)$.*

A partial proposal corresponds to a not final scenario. We consider the minimum among the possible outcomes, which means that we assign to a partial proposal a utility corresponding to the lower bound of the possible outcomes which can be achieved by accepting this proposal. For instance, if agent D is evaluating the partial proposal $\langle (e2, c2), (c1, b) \rangle$ from agent C , she attributes it the utility $\min(ut_D^{fw}(AF_3^+), ut_D^{fw}(AF_7^+)) = \min(0, -0.2) = -0.2$.

Definition 5 (Proposals evaluation). *Given a proposal $\langle P^+, P^- \rangle$ corresponding to the scenario $AF^+ = \langle A, R, P \rangle$, an agent ag_i associates it a utility $ut_i^{prop}(AF^+)$ which is $ut_i^{prop}(AF^+) = ut_i^{fw}(AF^+)$, if AF^+ is a final scenario, and $ut_i^{prop}(AF^+) = \arg \max_{AF_j^+ \in S_F(AF^+)} ut_i^{fw}(AF_j^+)$ otherwise.*

The GP (*Global Protocol*) we propose is inspired by the negotiation protocol with simultaneous answer proposed by Kraus [8], based on the model of alternate offers of Rubinstein [11]. It has the following features: the agents have no roles inside the negotiation protocol (differently from, for example, the Contract Net protocol [12]), there are no turns and the negotiation is managed synchronically. Kraus [8] presents a way to synchronize the counter-proposals while we extend this synchronization to each agent involved in the negotiation. GP is characterized by a number of rounds t_1, t_2, \dots . At each round t_j each agent ag_i puts forward a proposal p_i^j . If, at round t_j , a set of agents $AG_i \in AG$ arrives to a solution S with the resulting proposal p_{ris} , at round t_{j+1} each agent $ag_k \in AG_i$ can communicate the speech act $OK(p_{ris})$, accepting temporarily

the solution S . If every agent ag_k does it, S is called *temporal solution*. If, at round $t_w (w > j + 1)$, an agent ag_k communicates $OK(q_{ris})$, such as she accepts another solution, S is no more a temporal solution. Agent ag_k can communicate again $OK(p_{ris})$ in a round $t_z (z > w)$. A further speech act is introduced: NIL . An agent communicates NIL if she does not have further proposals to put forward. The speech acts are mutuized for Kraus [8] and Endriss [5], respectively. A negotiation ruled by the GP protocol terminates if either a temporal solution S is reached *s.t.* $dom(S) = off(S) = AG$ or a round t_j *s.t.* $\forall ag_i \in AG, p_i^j = NIL$ is reached. In the first case, the protocol terminates if all the agents involved in the negotiation find a common solution and accept it. In this case, the framework is modified coherently with the scenario which corresponds to the accepted solution. In the second case, the solutions which modify the framework are the temporal solutions of the negotiation.

The SP (*Server Protocol*) is defined in order to have a different testbed for our framework. It combines elements from the Contract Net protocol [12] and the protocol proposed by Kraus [8]. The features of this protocol are: an agent is chosen to have the role of the server (this choice can be taken by means of a voting phase or it can be driven by the features of the agents, e.g., the agent who controls more potential attacks), the other agents participate to the negotiation as clients, the server declares the *conflict deal*, such as the result assigned to the negotiation if it fails which consists, in our framework, in an activation/deactivation of a potential attack from the server.

Concerning termination, SP terminates when a solution is reached. Thus the speech act OK is not necessary and do not exist temporal solutions. The speech act NIL is admitted for clients. The server is allowed to use a particular speech act, END , which terminates the negotiation. A negotiation ruled by protocol SP terminates if either a solution is reached or the server communicates END or a round t_j *s.t.* $\forall ag_i \in AG \setminus \{SERVER\}, p_i^j = NIL$ is reached. In the first case, the protocol terminates if the server and a set (potentially empty) of agents find a common solution. The framework is modified coherently with the scenario corresponding to the accepted solution. The second and third cases correspond to a termination for conflict deal: in the second case, the failure is forced by the server while in the third case, the failure is caused by a lack of new proposals from the clients. In these cases, the framework is modified coherently with the conflict deal.

In the definition of the two protocols above, we have taken inspiration from different protocols existing in the literature: the notion of monotonic concession from [9] and [6], the roles of server and clients from [12], the speech acts from [7] and the notion of conflict deal from [5]. Our aim is to create two models for negotiation on potential attacks with substantial differences among them in order to test the performances of different strategies in different contexts.

We now present a number of reasoning mechanisms used by the agents to create the negotiation proposals and to decide whether to accept, refuse or raise other agents' proposals. Given a set of agents, each agent can implement different

strategies. We assume that the agents compute the utilities for every scenario before starting the negotiation process. We define the following strategies:

Designative agent: monotonic concession. The agent ignores the proposals of the other agents and, at each turn, she advances the more convenient proposal not yet advanced [5].

Simple reactive agent: conflict deal - aware. The agent bases her proposals on the conflict deal, she compares it with her actual proposal and she evaluates whether to advance new proposals or to communicate *NIL*. This strategy is based on the Zeuthen strategy.

Simple reactive agent: solution - aware. The agent considers the other agents' previous negotiation proposals and evaluates whether she can create a negotiation proposal which can be integrated with the others in order to create a solution; she compares that proposal with her actual proposal and she decides which one to advance.

Strategic agent: game theory. Since the outcomes of the other agents are not known, the Nash equilibrium cannot be computed with certainty. An agent who uses the notion of Nash equilibrium can represent the relation with another agent as a game in normal form expressing the utilities in function of a set of variables, corresponding to the other agent's parameters and then she can evaluate the existence of an equilibrium point for any or, at least, for one value assigned to the variables.

Strategic agent: do-ut-des. An agent adopting a do-ut-des strategy creates the proposals when she knows she can offer a precise exchange with another agent. If there exist different proposals with these requirements she advances the one with maximal utility.

4 Coalitions

Coalitions are modeled as a side-effect of the negotiation process and thus as a structure which is created in correspondence of the achievement of a solution during the negotiation. We introduce the notion of multiagent argumentation system *MAAS* and we define the notion of *power*.

Definition 6 (Multiagent argumentation system). *A multiagent argumentation system MAAS is a tuple $\langle AG, SC, AF^+ \rangle$ such that: AG is a set of agents ag_i with associated parameters α, β, γ and with their own negotiation strategies, SC is a coalition structure associated to AG , $AF^+ = \langle A, R, P \rangle$ is an extended argumentation framework linked to AG by relation R_{app} .*

Definition 7 (Agents and coalitions power). *Let $MAAS = \langle AG, SC, AF^+ \rangle$:*

- val_i of an agent ag_i is $| a \in A \text{ s.t. } R_{app}(a, ag_i) \wedge IN(a) |$,
- val_j of a coalition c_j is $\sum_{ag_k \in c_j} val_k$,
- val_{SC} of a coalition structure is $\sum_{c_j \in SC} val_j$.

- $pow_i^{AG} : AG \longrightarrow \mathbb{R}$, $pow_i^{AG}(ag_i) = \frac{val_i}{val_j}$ for $ag_i \in c_j$
- $pow_j^{SC} : SC \longrightarrow \mathbb{R}$, $pow_j^{SC}(c_j) = \frac{val_j}{val_{SC}}$ for $c_j \in SC$

The value of an agent is a natural number which indicates the number of accepted arguments the agent owns and the value of a coalition is the sum of the values of the single agents composing it. The value of a coalition structure is the sum of the values of the single coalitions composing it. The power of an agent is the tradeoff between her value and the value of her coalition. The power of a coalition is the tradeoff between its value and the value of the coalition structure it belongs to. These values are assigned in order to manage the equilibria of the agents inside the coalitions: we attribute more influence in making decisions to the more powerful agents.

Algorithm 1. Intra-coalition negotiation

```

1: Each agent  $ag_i \in c_j$  proposes  $p_i$  following her private strategies;
2: if one of the proposals is END then
3:   Categorical voting on term;
4:   Eventual split of the coalition in  $c_{YES}$  and  $c_{NO}$ ;
5:    $c_{YES}$  communicates END,  $c_{NO}$  communicates NIL;
6: end if
7: if If one of the proposals is OK( $p$ ) then
8:   Preferential voting among term and  $p$ ;
9:   if the result is  $p$  then
10:    The coalition communicates OK( $p$ );
11:   end if
12:   if the result is term then
13:    Instances of OK( $p$ ) are removed from the set of proposals of the agents and go back to
14:    (7);
15:   end if
16: if all the proposals are scenarios or NIL then
17:   Verify the compatibility of the set of proposed scenarios
18:   if compatibility holds then
19:    Compute the natural mediation among the proposed scenarios;
20:   else
21:    Compute deconflictualized mediation;
22:   end if
23:   Categorical voting on the mediation;
24:   Eventual split of the coalition in  $c_{YES}$  and  $c_{NO}$ ;
25:    $c_{YES}$  e  $c_{NO}$  restarts independently from (16);
26: end if
    
```

At the beginning of the negotiation process each agent belongs to a singleton coalition. Afterwards, a coalition is created in correspondence with the resolution of a negotiation phase. The new coalition involves those coalitions among which the solution of the negotiation has been found. If the negotiation terminates in conflict deal, given that the agents do not have reached an agreement, no coalition is formed. This choice derives from the decision to model the creation of a coalition structure as a parallel process to negotiation and derived from it. Coalition formation is equivalent to the definition of agreement in negotiation. Similar motivations guide the proposal of Cohen and Levesque concerning the notion of *shared intention* [3]: two agents have the shared intention to perform a

collective action if each agent intends to do part of the work in order to achieve the action. On the contrary, a coalition is split if the members composing it do not find an agreement on the negotiation proposal of the other coalitions.

Different intra-coalition mechanisms can be defined and based on the concepts of power and relative sociality. Due to lack of space we omit formal definition and formulae for the next concepts:

- A **natural mediation** is the resulting proposal from a consistent set of negotiation proposals.
- A **deconflictualized mediation** is the resulting proposal from an inconsistent set of negotiation proposals.
- A **preferential voting** asks each agent to express preference over a certain negotiation proposal.
- A **categorical voting** asks each agent to accept or refuse a specific negotiation proposal.

These elements are used in the intra-coalition negotiation Algorithm 1.

5 Experiments

The protocols and strategies introduced in the previous sections have been implemented in Java in order to compare the negotiation results in different test environments. We focus on two groups of tests: those based on homogeneous multiagent systems (a set of agents implementing the same strategy) and those involving heterogeneous systems for comparing strategies.

Table 2. Numerical results

strategy	rounds	% in	% act	% coal	VS	mono	cdaw	solaw	dodes	nash
mono	2.1	0.748	0.269	0.735	mono	/	+8	-4	+8	+6
cdaw	3.025	0.743	0.275	0.903	cdaw	+8	/	-6	+6	+6
solaw	2.1	0.749	0.268	0.735	solaw	-12	+3	/	+8	+6
dodes	2.45	0.663	0.739	0.815	dodes	-4	+3	-2	/	+4
nash	2.55	0.698	0.500	0.789	nash	+2	+6	+2	+4	/

In the first class of tests, for each combination of MAAS, strategy and negotiation protocol a test has been run. Table 2.a reports highly aggregated data for this class of tests: the number of rounds needed by the agents to reach a solution, the percentage of accepted arguments in the final scenario, the percentage of activated potential attacks and, when dealing with coalitions, a numerical value describing the coalition structure: it is a value spanning from 0 to 1, where 0 means *collection of singleton coalitions* and 1 *grand coalition*.

The second tests' group aims at comparing different strategies. Our inspiration was Axelrod's Iterated Prisoner Dilemma's Tournament . For each protocol/MAAS/pair of strategies, we split the agent set in half and we assign to

each half a different strategy. Then we run a dual test after switching the same two strategies assigned to the same two agents' subsets. Our goal is to check if a strategy could grant better results independently of which subset of agents is implementing it. We base our evaluation over the number of accepted arguments owned by each subset of agents at the end of the negotiation phase: the possess of a higher number of accepted arguments reflects how well a strategy works in a certain protocol and MAAS. A single example of comparison among strategies is depicted in Table 2.b.

Results show that the maximality component in the solution concepts' definition causes agreement among broad sets of agents and short negotiations: the average number of rounds is below 3 and almost always a single, big coalition emerges from the negotiation process, leaving a small collection of single coalitions out. Due to the low number of rounds, the choice of the negotiation protocol does not impact much. The enumerative strategies are the best performing, although they obviously require more computational time, since they generate and examine an exponential number of scenarios. The do-ut-des and game-theoretical strategies cause less-competitive results, due to the clash between their local approach and the global one of argumentation semantics.

6 Conclusion

We start from the the work proposed by Boella *et al.* [2], where an integration between the extended argumentation framework and a multiagent system, based on partitioning of arguments over agents and introducing opaque utility functions is presented. We enrich it by introducing two negotiation protocols based on a set of negotiation strategies and by testing the properties and interactions of these components. Although negotiation and argumentation are widely discussed in the literature (e.g. [1]), we claim the originality of our approach, where argumentation is not used as a negotiation technique but as a modeling tool which becomes the object of the negotiation process. Simari *et al.* [10] propose a dynamic argumentation framework, i.e. an extended argumentation framework from which different Dung-style frameworks can be extracted. The two approaches differ, because we use abstract argumentation while Simari does not, and in our case, each agent owns all the information about her arguments, and she has to decide whether to communicate it or not, while in [10] the resulting framework derives from a process of evidence checking rather than deliberation, social choices and strategic issues. Future work addresses the introduction of new non-enumerative (polynomial) strategies as well as coalition - aware ones and the study of their impact.

References

1. Amgoud, L., Dimopoulos, Y., Moraitis, P.: A unified and general framework for argumentation-based negotiation. In: Procs. of the 6th Int. Conf. on Autonomous agents and multiagent systems (AAMAS), pp. 963–970. ACM, New York (2007)

2. Boella, G., Gabbay, D.M., Perotti, A., Villata, S.: Coalition formation via negotiation in multiagent systems with voluntary attacks. In: *Procs. of the 22nd Benelux Conf. on Artificial Intelligence (BNAIC)*, pp. 25–32 (2010)
3. Cohen, P.R., Levesque, H.J.: Teamwork. *Nous.* 25(4), 487–512 (1991)
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2), 321–358 (1995)
5. Endriss, U.: Monotonic concession protocols for multilateral negotiation. In: *Procs. of the 6th Int. Conf. on Autonomous agents and multiagent systems (AAMAS)*, pp. 392–399 (2006)
6. Gmytrasiewicz, P.J.: A review of rules of encounter: Designing conventions for automated negotiation. *AI Magazine* 16(2), 83–84 (1995)
7. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. *Humanities, Social Sciences and Law* 10, 199–215 (2000)
8. Kraus, S.: Automated negotiation and decision making in multiagent environments. In: *Procs. of MultiAgent Systems and Applications (ACAI-EASSS)*, pp. 150–172. Springer, Heidelberg (2001)
9. Rosenschein, J.S., Zlotkin, G.: *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge (1994)
10. Rotstein, N.D., Moguillansky, M.O., García, A.J., Simari, G.R.: A dynamic argumentation framework. In: *Procs. of the Int. Conf. on Computational Models of Argument (COMMA)*, pp. 427–438 (2010)
11. Rubinstein, A.: Perfect equilibrium in a bargaining model. *Econometrica* 50 (1987)
12. Smith, R.G.: The contract net: A formalism for the control of distributed problem solving. In: *Procs. of the 5th Int. Conf. on Artificial Intelligence (IJCAI)*, p. 472 (1977)

Environment-Driven Threats Elicitation for Web Applications

Hui Guan^{1,3}, Weiru Chen¹, Lin Liu², and Hongji Yang³

¹ Shenyang University of Chemical Technology

guanh1999@126.com, willc@china.com

² School of Software, Tsinghua University, Beijing, China

linliu@tsinghua.edu.cn

³ Software Technology Research Laboratory, De Montfort University, Leicester, England

hyang@dmu.ac.uk

Abstract. The popularity and complexity of web application present challenges to the security implementation for web engineering. Threat elicitation is an indispensable step for developers to identify the possible threats to the web applications in the early phase of software development. In this context, a novel approach is proposed to ease the threats elicitation for web application by using a defined web application classification as the sieve to sift a common threat list. The final result shows that the proposed model is a simplified and effective solution to threats elicitation to web application.

Keywords: web application, threat, security, classification, risk.

1 Introduction

The ease of implementation and use of web technologies has made them an omnipresent and essential component of online commercial sites, intranet and extranet application, as well as the internet services offered and used by companies, which creates new challenges for the web engineering. This is because web applications open systems and information to be accessed by public. In this type of software, requirements have become more complex, in order to guarantee information security [1]. Comprised web server can damage organizations in many ways, from surrendering customer privacy data and accepting fraudulent transactions to indirectly damaging corporate reputation as the result of defaced homepage. Report has shown that web applications account for 80 percent of internet vulnerabilities in the second half of 2008 and rose in prevalence by about eight percent from the first half of the year [2]. Security in web engineering has become an emergent task.

Given the increasing complexity and frequency of web application risks, decision makers must take action to sufficiently protect the web applications. However, security as an architectural driver is often at the expense of performance (e.g. component redundancy), usability (e.g. complexity of using the application) and cost (e.g. using SSL to implement HTTP requires PKI or third party certificates, slows traffic, etc.). Most development companies are having a tough time balancing all of these factors [3].

Threat modeling for web application is one of the effective methods to figure out the threat to the web application security at the early stage. This paper depicts a new method to ease eliciting threats to web application by using web application classification which is proposed taking consideration of security-related environmental factors. The remainder part of this paper is arranged as follows. Section 2 illustrates our proposed model and its detailed context, while Section 3 elaborates the proposed approach by a case study and results have been discussed. Section 4 reviews the related works in this area. Finally, Section 5 concludes by summarizing the key contributions and outlining the future steps.

2 The Proposed Approach

Web application is some kind of different one due to its application environment and complexity. Accordingly, threats to this kind of software are different to some extent. The number and category of the threats to different kind of web applications may not be the same when taking account of complexity and environment where the applications are hosted. Sometimes, the applications may be developed with security in mind, and may be difficult to penetrate as well, but if the environment where the application is hosted is not properly secured, it is easy to penetrate the environment, and as a result, it is easy to compromise the whole application including its subsystems and platform.

As far as the web application developers are concerned, on one hand, they need to keep security on mind when developing the web application, on the other hand, they are usually forced to face the dilemma that how to trade-off among so many product factors such as security requirements, product deadline and budgets etc. Thus, this paper proposes a different approach to ease the elicitation of the threats for web applications by defining web application classification as the filter to rule some threats out immediately according to the security requirement and the given scenery. Fig.1 gives an overall idea of this model.

The proposed model is working as a sieve to sift the inappropriate threats. With this approach, we start with a laundry list of common threats [12] grouped by network, host, and application categories. Next, apply the threat list to the given application architecture and screen out the threats matching its own web application category. Then, further filtering can be done to the result threat set according to the security requirements of the given web application. We will be able to rule some threats out because they do not apply to the scenario of the given application. As a result, a set of filtered threats specific to the given web application can be obtained.

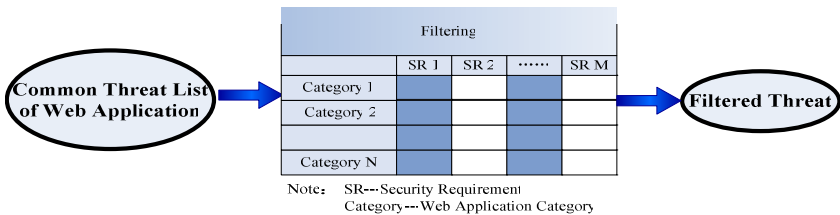


Fig. 1. Illustration of the Proposed Approach

2.1 Classification of Web Application

All web-applications are not same, the architecture and its supporting systems could be different for each application depending on the complexity, but the resources or techniques needed for running those applications may be same. So, it is possible to create a common threat model and identify all possible threats that can be used for all web-applications [17]. The Web Application Classification is a cooperative effort to clarify and organize the types of web application with different security risk level so as to ease the process of threats identification. It helps you to identify which threats are relevant to your application through the proposed model.

As has discussed in Section 1, the risks web application faced are certain to be different when they are hosted in well secured and non-secure environments. Thus, it is necessary for us to take consideration of the web application type before further discussion. For the consideration of the environment where web application hosted, three attributes of web applications have been taken into account, which are the usage scope, target user and connectivity mode. Then, the circumstances of all the attributes of web application are listed in Table 1.

Table 1. Attributes of Web Application

Usage Scope (US)	Target User (TU)	Connection Mode (CM)
Internal use only (US1)	Know users (TU1)	Intranet (CM1)
Blended use (US2)	Anonymous (TU2)	VPN (CM2)
External use only (US3)		Internet (CM3)

The number of web application types abbreviated as WA equals to:

$$WA=US \times TU \times CM \quad (1)$$

There should be 18 web application types according to the formula 1. However, some of them are not applicable in real web application which should be ignored. For example, the combinations of (US1, TU1, CM3) or (US1, TU2, CM3) are not incompatible. All of the applicable web application types are listed in Table 2.

2.2 Threat Classification

This section we use web application classification to filter the threats proposed in [12] according to the given security requirements. In order to illustrate the detailed the steps of our approach, threats list used as the threats set to be filtered is described below.

For comprehensiveness, we choose the threat set proposed by [12] which enumerates the top threats that affect web applications at the network, host, and application levels¹. For the sake of being used conveniently in our approach, two related factors, web application category and CIA requirement are added in Table 3, Table 4 and Table 5.

¹ For paper space limitation, we don't give the detail description of each threat, for further information, please see [12].

Table 2. Web Application Classification

ID	Name	Attribute	Definition	Security risks
WA1	Internal use facing known users via intranet	US1 TU1 CM1	Application used primarily on the internal network of an organization for a mount of known users.	This kind of applications is designed for internal use so that only internal users can access from intranet. Therefore, the security risk is considered as low.
WA2	Internal blended External use facing known users via VPN	US2 TU1 CM2	Application used primarily on the internal network, but a mount of known external clients can access through VPN	The security risk is low but there are possibilities for sharing user-credentials, impersonation and sniffing on the external client site.
WA3	External use facing known users via Internet	US3 TU1 CM3	Application used for external use. A mount of known users can access from internet	The security risk is a bit higher compared to previous types because it is exposed to all kinds of attacks from internet, however, it is not very high for only known users can access
WA4	External use facing public users via Internet	US3 TU2 CM3	Application used for external use. Public users can access from internet	The security risks of these applications are considered little bit high compared to previous types since they are open to public from Internet
WA5	Internal blended external use facing known users via Internet	US2 TU1 CM3	Application used for internal users and external known users from Internet	The security risks of these applications are higher due to their design complexities. Usually, this kind of applications are designed primarily for internal use, it is a little more dangerous when known users access from Internet
WA6	Internal blended external use facing public users via Internet	US2 TU2 CM3	Application used for internal users and external public users from Internet	The security risks of these applications are highest due to their design complexities. Usually, this kind of applications are designed primarily for internal use, it is the most dangerous when public users access from Internet due to lack of security controls

Table 3. Network level threat [12]

No.	Threat Name	WA Category	CIA Risk
1	Information Gathering	WA3 WA4 WA5 WA6	C
2	Sniffing	WA3 WA4 WA5 WA6	C
4	Spoofing	WA3 WA4 WA5 WA6	CIA
5	Session Hijacking	WA3 WA4 WA5 WA6	CIA
6	Denial of Service	WA3 WA4 WA5 WA6	A

Table 4. Host level threat [12]

No.	Threat Name and Description	WA Category	CIA Risk
7	Viruses, Trojan horses, and Worms	ALL	CIA
8	Footprinting	WA4 WA6	C
9	Password Cracking	ALL	CI
10	Denial of Service	ALL	A
11	Arbitrary Code Execution	WA3 WA4 WA5 WA6	CIA
12	Unauthorized Access	ALL	CIA

Table 5. Application level threat by application vulnerability category [12]

No.	Threat Name and Description	WA Category	CIA
Input Validation			
13	Buffer Overflow	WA3WA4 WA5 WA6	CIA
14	Cross-Site Scripting (XSS)	WA4WA6	CIA
15	SQL Injection	WA3WA4 WA5 WA6	CIA
16	Canonicalization	WA4WA6	CIA
Authentication			
17	Network Eavesdropping	WA3WA4 WA5 WA6	C
18	Brute Force Attacks	WA4 WA6	CIA
19	Dictionary Attacks	WA4 WA6	CIA
20	Cookie Replay	WA3WA4 WA5 WA6	CIA
21	Credential Theft	ALL	CIA
Authorization			
22	Elevation of Privilege	ALL	CIA
23	Disclosure of Confidential Data	ALL	C
24	Data Tampering	ALL	IA
25	Luring Attacks	WA3WA4 WA5 WA6	CIA
Configuration Management			
26	Unauthorized Access to Administration Interfaces	WA4WA6	CIA
27	Unauthorized Access to Configuration Stores	WA4 WA6	CIA
28	Retrieval of Clear Text Configuration Data	WA3WA4 WA5 WA6	C
29	Lack of Individual Accountability	WA3WA4 WA5 WA6	CIA
30	Over-Privileged Process and Service Accounts	WA4WA6	CIA
Sensitive Data			
31	Access sensitive data in storage	ALL	CI
32	Network Eavesdropping	WA4 WA6	C
33	Data Tampering	ALL	I
Session Management			
34	Session Hijacking	WA4 WA6	C
35	Session Replay	WA4 WA6	CIA
36	Man in the Middle	ALL	C
Cryptography			
37	Poor Key Generation or Key Management	WA3WA4 WA5 WA6	C
38	Weak or Custom Encryption	WA3WA4 WA5 WA6	C
39	Checksum Spoofing	WA3WA4 WA5 WA6	CI
Parameter Manipulation			
40	Query String Manipulation	WA3WA4 WA5 WA6	CA
41	Form Field Manipulation	WA4 WA6	CA
42	Cookie Manipulation	WA4 WA6	CI
43	HTTP Header Manipulation	WA3WA4 WA5 WA6	IA
Exception Management			
44	Attacker Reveals Implementation Details	WA3WA4 WA5 WA6	C
45	Denial of Service	WA3WA4 WA5 WA6	A
Auditing and Logging			
46	User Denies Performing an Operation	ALL	CI
47	Attacker Exploits an Application Without Trace	ALL	CI
48	Attacker Covers His or Her Tracks	ALL	CI

2.3 Algorithm

In this section, the process of filtering threats from common threat list according to its web application type and security requirements is described in the following algorithm. Starting from the web application classification, each threat in common threat list is sieved by the rule, as a result, a threat list applying for the given web application can be obtained. Just like described in [12], “a threat is any potential occurrence, malicious or otherwise, that could harm an asset. In other words, a threat is any bad thing that can happen to your assets”. It is meaningless to discuss threats without connection to their assets. Hence, it is necessary to associate the threats to their comprised assets so that web application developer can design proper security mechanism to protect the assets.

Algorithm 1

Step 1: Classification

Classify the given web application into one of the proposed web application type according to three attributes, use W_{Ai} to represent

Step 2: Rating the security requirements CIA of W_{Ai}

Web application is rated “Low”, “Medium”, or “High” on the metrics of Integrity, Availability, and Confidentiality, use {CIA requirements} to represent

Step 3: Filtering

for all threats T_i in common threat list CTL do /* T_i stands for the i^{th} threat*/

if T_i . WA Category == All then

*$T_i \rightarrow \{TL\}$ /*Insert T_i to threat list TL */*

end if

if W_{Ai} . WA Category $\in T_i$. W_{Atype} then

*$T_i \rightarrow \{TL\}$ /*Insert T_i to TL */*

end if

end for

Step 4: Further Filtering

for all TL_i in TL do

if TL_i . CIA risk does not match the {CIA requirements} then

*$TL_i \leftarrow \{TL\}$ /*Remove TL_i from TL */*

end if /* C: Confidentiality, I: Integrity, A: Availability*/

end for

Step 5: Connection

Establish the connection of each filtered threat to its target assets

3 Case Study

In order to illustrate the usefulness of web application classification this paper will examine a case study used in [3]. Widgets Incorporated is a medium-sized consumer goods company. They have determined the need to create I-Tracker: a custom-built inventory tracking application to facilitate growing customer demand. The most common use case will be for sales staff to enter data from a sales order which will automatically update the inventory levels and alert the logistics staff to prepare the order for shipment. When the inventory level for a particular widget drops below a certain threshold the manufacturing division will be notified. The main types of data

used in the application include inventory levels, customer IDs, sales orders numbers, descriptions of orders, and product IDs.

Step1: Classification. From the case description, the example web application is used within internal environment facing a number of known users. Thus, it belongs to the first type of web application categories.

Step2: Rating security requirements CIA. Using internal guidelines based on documents such as [18], the following application classification may be produced:

- Confidentiality: Low. All data in the application is readily available to anyone in the company. Sensitive financial data and client private information are not handled by this application.
- Integrity: High. Poor inventory and shipping tracking may result in significant financial loss to the company and customer dissatisfaction or loss of customers.
- Availability: Medium. A major disruption of the application will cause a backlog in shipping and have some financial consequences to the organization. Minor disruptions, however, can be tolerated as customers expect a 4-6 week delay in receiving their goods.

Step3: Filtering. According to the algorithm 1 described in Section 2.3, the most likely threats are filtered and listed in Table 6.

Table 6. Threat list after filtering

ID	Threat Name	CIA
7	Viruses, Trojan horses, and Worms	CIA
9	Password Cracking	CI
10	Denial of Service	A
12	Unauthorized Access	CIA
21	Credential Theft	CIA
22	Elevation of Privilege	CIA
23	Disclosure of Confidential Data	C
24	Data Tampering	CI
31	Access sensitive data in storage	C
36	Man in the Middle	C
46	User Denies Performing an Operation	CI
47	Attacker Exploits an Application Without Trace	CI
48	Attacker Covers His or Her Tracks	CI

Step4: Further Filtering. The threat list can be further screened out according to the security requirements of CIA aspects. In terms of the algorithm described in Section 3, threats with only Confidentiality requirements can be rule out in that the given application has low requirements on Confidentiality, while threats with Integrity and Availability are remained.

Step5: Connection. In order to establish the connection of each filtered threat to its target assets, data flow diagram is a useful tool to identify the key assets. Data flow diagram of the example web application is listed in Fig. 2.

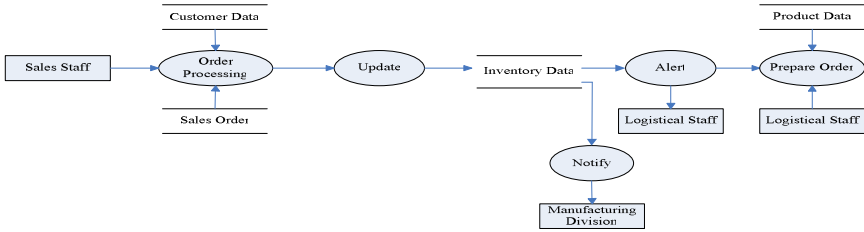


Fig. 2. Data flow diagram of the I-Tracker

Hernan et al. [19] proposed that elements in data flow are vulnerable to different kinds of threats, which is listed in Table 7.

Table 7. Threats and Its Possible Target Assets

ID	Threat Name	Assets
7	Viruses, Trojan horses, and Worms	Data Store, Data Flow
9	Password Cracking	Process, Data Store, Data Flow
10	Denial of Service	Process, Data Store, Data Flow
12	Unauthorized Access	Data Store, Data Flow
21	Credential Theft	Process, Data Store, Data Flow
22	Elevation of Privilege	Processes
24	Data Tampering	Process, Data Store, Data Flow
46	User Denies Performing an Operation	Processes, External Entity
47	Attacker Exploits an Application Without Trace	Data Store, External Entity
48	Attacker Covers His or Her Tracks	Data Store, External Entity

4 Related Works

Threat elicitation is within the domain of software security requirement. In our earlier work, we focus on the security design using social modeling concept [4] and a systematic evaluation of security requirements is proposed [5]. Social modeling concept is used to analyze the business and organizational context of systems with regard to security [4]. A 3-dimensional vector for quantitative evaluation of security requirements has been proposed, which takes into account the importance of assets to be protected, the vulnerability of the system and the trustworthiness of environment [5].

In terms of threat elicitation, several methods have been proposed. One type of proposed method is abuse cases, misuse cases [6, 7, 8] which can be used once the system use cases have been created. Threat can be elicited by analyzing abuse and misuse cases [6]. [8] extends the misuse cases to misuse activities which are analyzed to see how it could be subverted to produce a misuse of information, as a result, a set of threats can be listed. However, the practical method for creating misuse cases or activities is usually with the process of brainstorming [6].

The threat modeling approach [9, 10, 11] is another approach which gives a clear idea of how to elicit, classify, prioritize and mitigate threats. Especially, some threat modeling approaches are designed for web applications [12, 13]. Tøndel et al. [10]

shows that threat modeling often is considered as an important part of the requirements phase, as well as an iterative process, continuously revisited throughout the software lifecycle. Oladimeji et al. [11] propose a goal-oriented approach to threat modeling where the notions of negative soft-goals are used for representing threats. In [12] a comprehensive approach is provided to building highly secure and feature-rich web applications using the .NET Framework. [13] elaborates, illustrates and discusses the threat modeling process and its usefulness to the architectural designs of an e-banking application. Although threat modeling is seen as a thorough approach to threat elicitation, mitigation and management, however, it demands information available only at late design time which drives the security design to start in the middle of the software development life cycle.

Another method to elicit the threat is problem frame [14, 15]. A problem frame characterizes a class of simple problem. Realistic problems are seen as compositions of simple problems of recognized to elicit and analyze software security requirements [15, 16]. Hatebur et al. [16] describe a security engineering process to develop security systems based on problem frames, and a collection of security patterns, plus components as the way to deal with the solution. While problem frames appear useful in some cases, they are not as useful for a complete design as UML. Also, they are not so widespread.

5 Conclusion

Security problems in web application have become an emergent task for developer to solve. Threat elicitation is an effective method to identify the possible threat to the application in the early stage of development. This paper proposes a novel approach to ease the elicitation of threats for the given application with the aid of web application classification which is defined taking consideration of the complexity and environments where the web applications are hosted. With the proposed method and result obtained under this work, it is possible to determine the appropriate design to help identify the most critical threats in the web application.

One defect of our proposed model is that it is incapable of identify the emerging threats outside the common threat list. Fortunately, the common threat list can be extended along with the emergency of the new threats. Another defect is that the filtered threats by the proposed model are not as specific as elicited by other methods. However, our proposed method can shorten the threat elicitation time significantly and an experienced developer can easily relate the filtered threats to its appropriate scenery. Future research will focus on improve the proposed approach to an automatic implementation and integrate it into the security risk assessment for the web application.

Acknowledgment

This work was sponsored by Liaoning Province Office of Education of China Project Research on Security-oriented Software Reengineering (Grant No. L2010439) and partial financial support by the National Natural Science Foundation of China (Grant No. 60873064 and Grant No. 90818026).

References

1. Brunil, R., Hisham, M.H., Jorge, M.: A methodological tool for asset identification in web applications. In: 2009 Fourth International Conference on Software Engineering Advances, pp. 413–418 (2009)
2. Web application security trends report, http://www.cenzic.com/downloads/Cenzic_AppSecTrends_Q3-Q4-2008.pdf
3. Sethi R.: The Importance of application classification in secure application development, version 1.0 (2007), <http://www.webappsec.org/projects/articles/041607.shtml>
4. Liu, L., Yu, E., Mylopoulos, J.: Secure Design Based on Social Modeling. COMPSAC 2006: Proceedings of the 30th Annual International Computer Software & Applications Conference, Chicago, Illinois, USA, pp. 71–76 (2006)
5. Long, T., Liu, L., Yu, Y., Jin, Z.: AVT Vector: A Quantitative Security Requirements Evaluation Approach based on Assets, Vulnerabilities and Trustworthiness of Environment. In: Proceedings of 17th IEEE International Requirements Engineering Conference, Atlanta, USA (2009)
6. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements Engineering 10, 34–44 (2005)
7. Alexander, I.: Misuse Cases: Use Cases with Hostile Intent. IEEE Software 20(1), 58–66 (2003)
8. Braz, F.A., Fernandez, E.B., VanHilst, M.: Eliciting security requirements through misuse activities. In: 19th International Conference on Database and Expert Systems Application, pp. 328–333 (2008)
9. Swiderski, F., Snyder, W.: Threat modeling. Microsoft Press (July 2004)
10. Tøndel, I.A., Jaatun, M.G., Meland, P.H.: Security requirements for the rest of us: a survey. To appear in IEEE Software (January/February 2008)
11. Oladimeji, E., Supakkul, S., Chung, L.: Security threat modeling: a goal-oriented approach. In: Proceedings of SEA 2006, Dallas, TX (2006)
12. Curphey M., Scambra J., Olson E.: Improving web application security: threats and countermeasures (2003), http://www.cgisecurity.com/lib/Threats_Countermeasures.pdf
13. Möckel, C., Abdallah, A.E.: Threat modeling approaches and tools for securing architectural designs of an e-banking application. In: 6th International Conference on Information Assurance and Security, art. no. 5604049, pp. 149–154 (2010)
14. Jackson, M.: Problem Frames, Problem frames and software engineering. Expert Systems 25(1), 7–8 (2008)
15. Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: a framework for representation and analysis. IEEE Trans. on Software Engineering 34(1), 133–153 (2008)
16. Hatebur, D., Heisel, M., Schmidt, H.: Analysis and component-based realization of security requirements. In: Proceedings of the International Conference on Availability, Reliability and Security (AREs), pp. 195–203 (2008); IEEE Transactions
17. Jesan J. P.: Threat modeling web-applications using STRIDE average model. In: Computer Security Conference (2008)
18. Director of Central Intelligence Directive 6/3: Protecting sensitive compartmented information within information systems, http://www.fas.org/irp/offdocs/DCID_6-3_20Manual.htm
19. Hernan S., Lambert S., Ostwald T., Shostack A.: Uncover security design flaws using the STRIDE approach (2006), <http://msdn.microsoft.com/en-us/magazine/cc163519.aspx>

Towards Ontology of Fraudulent Disbursement

Czesław Jędrzejek, Jolanta Cybulka, and Jarosław Bąk

Institute of Control and Information Engineering, Poznań University of Technology
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
{czeslaw.jedrzejek,jolanta.cybulka,jaroslaw.bak}@put.poznan.pl

Abstract. The task to ontologically model the knowledge concerning the selected class of economic crimes is considered; particularly we focus on fraudulent disbursement. The ontology has a layered structure with the foundational ontology (constructive descriptions and situations) on the structure's top and the application ontology at the structure's bottom. The application level entities were manually separated from the motivating crime scenarios, having a domain- and a task-based parts. Domain-based ontology contains descriptions of attributes and relations of the domain while the task-based part, designed to support the knowledge extraction from databases, is implemented via rules that are used to extract data about documents and their attributes, transactions, engaged people actions and their legal qualifications.

Keywords: Financial crime, fraudulent disbursement ontology, legal sanctions assignment.

1 Introduction

Economic crimes are particularly difficult to model because fraudsters use many types of criminal activity schemes, thus, it seems impossible to construct a simple conceptual model of any generality. Here, we present an ontology-based model of a simple economic crime, namely *causing damage to a company* (in Polish: działanie na szkodę spółki), in particular the asset misappropriation that includes the fraudulent disbursement. Such a crime is very widespread and intractable across countries and industries. From the 2009 year survey [13] it followed that asset misappropriation constituted two-thirds of all economic crimes, often accompanied by money laundering.

In the previous work we modeled a simplified version of this kind of crime, called *Hydra case* [10]. Now we build an ontology that covers most of the crime but without the money laundering thread. Let us give a brief description of the considered crime. The (whitened) facts of the economic crime considered here were initially written on 7 pages in a textual form [14] by a public prosecutor and then processed by a knowledge engineer [12], to obtain a description in a natural-like language (with sentence schemes like "Company C1 hires company C2 to do work W at location L."). On the basis of facts the crime scheme in Fig. 1 was created, which illustrates acting agents (companies and persons) and

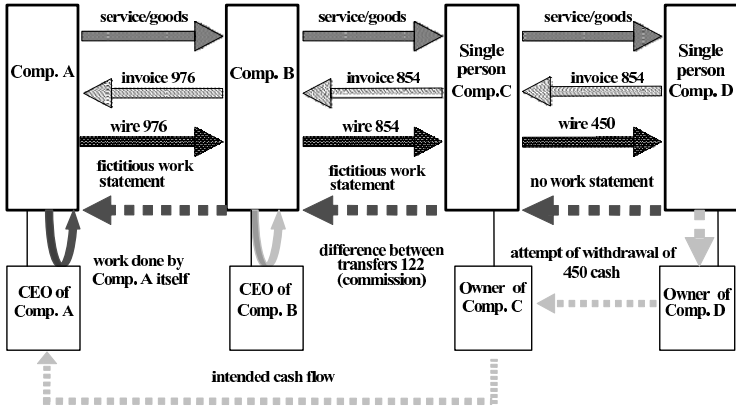


Fig. 1. A scheme of agents and activities in the Hydra case

chains of their activities. These activities are divided into 4 types of flows of: services/goods (here, the construction work), invoices accompanied by money amounts, wired money amounts and authorized legal documents concerning the services/goods. In the real case, fitting to this scheme, the *chief executive officer*, CEO, of company A (Hydra) subcontracted a construction work that had been then consecutively subcontracted through a chain of phony companies (B, C, and D). Each company falsified documents stating that the contracted work had been done. Actually, what was to be done as a "subcontracted construction work" company A did itself. At the end of the chain, the owner of a single person company D attempted to withdraw cash, and there was a suspicion that this cash had reached the management of company A "under the table" ("intended cash flow" in Fig.1). In [10] we assumed that only a single level of document's authorization existed: once an approval of the construction job was made, the payment for this work followed without further authorization. Consequently, only one person in company A was responsible for the crime. Here, we consider the three-level (real) structure of authorization of documents (see the rules in section 4). The created layered ontological structure (see Fig.2) has the foundational level on the structure's top, which is specialized by the core and then the application levels. The foundational level semantic entities were identified with the use of the content design pattern, namely the constructive descriptions and situations ontology (see section 2). The core level ontology contains the general entities connected with economic crimes, investigation and inquiry procedures and the dynamics of these processes [4]. The application level entities were manually separated from the motivating crime scenario with the requirement of identifying only necessary semantic entities that follow the logical order of uncovering a crime. As usual, this level is divided into two parts: domain-based and task-based. The former concerns the conceptualization of "a domain" whose attributes and relations are of interest while the latter supports the realization of the functionality of some application ("a task"). The task-based part is designed to support the

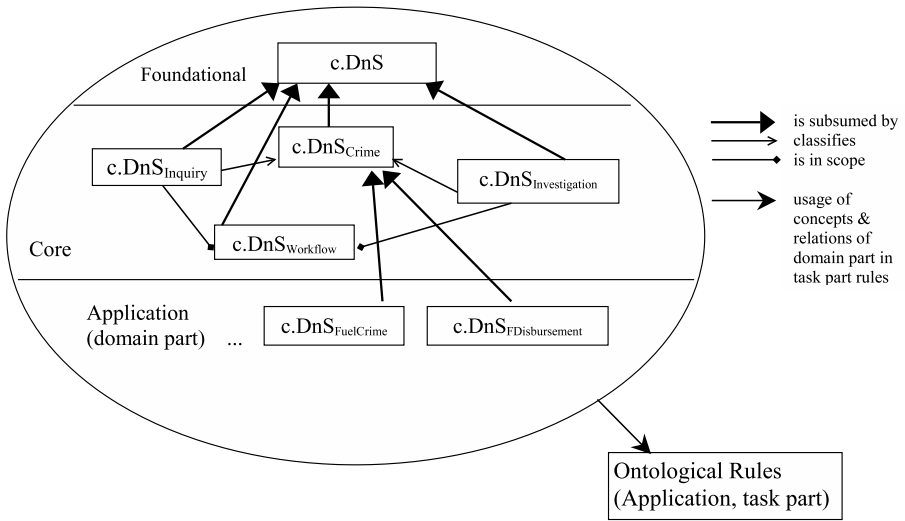


Fig. 2. A scheme of the fraudulent disbursement ontology

knowledge extraction from databases, thus, it is implemented via logical rules: they enable to extract suitable documents and their attributes, formal hierarchy in a company, parameters of transactions, engaged people actions and their legal qualifications. Section 2 presents some notes on ontology modeling methodology. Section 3 describes the domain-based part of fraudulent disbursement's, while section 4 contains the rules of task-based part of our model. Conclusions and future work are presented in section 5.

2 Notes on Ontology Design Methodology

There exist different methodologies of ontology engineering: TOVE, Enterprise Ontology, KACTUS, SENSUS, METHONTOLOGY, On-To-Knowledge, DILLIGENT, DOGMA, UPON, or the proposal of eXtreme ontology Design (XD) relying on pattern-based technique implemented in C-ODO ontology (Collaborative Ontology Design Ontology [8], [9]). The latter is worth noticing because the elements of it are adopted in our work. The NeOn methodology suggests formulating the specification of an ontology by expressing: 1) purpose of the ontology, 2) scope of the ontology, 3) level of formality, 4) intended users of the ontology, 5) its intended uses ("functionality"), 6) requirements specification (in the form of competency questions and answers), 7) pre-glossary of concepts and individuals.

We employed the c.DnS content pattern [7], actually its Polish language version. This assumption entailed the strategy of identification of concepts – it was a "top-down" approach. The c.DnS layer contains the most general categories of concepts that are related by general formal relationships. Its advantages as

follows: 1) its expressive power is very high and therefore it may be easily projected on different subdomains, 2) it is well suited to model domains populated with many non-physical entities, 3) it enables the addition of contextual or epistemic knowledge to the ontology, 4) the simplified version of the ontology expressed in the Web ontology language can be freely accessed [6].

In our view on economic crimes' reality the four different subdomains of interest exist (we call them "processes") that are related to each other (the core layer in Fig.2): a crime, an inquiry (in general) is carried by a public prosecutor with the help of police, an investigation, carried by the police and a workflow, which represents the dynamics of other processes. The ontological model composes them and each process is formalized by means of a specialized version of the *c.DnS* tuple, namely:

- *c.DnSForACriminalProcess*
- *c.DnSForAnInquiryProcess*
- *c.DnSForAnInvestigationProcess*
- *c.DnSForAWorkflowProcess*.

The fraudulent disbursement economic crime model is presented in sections 3 and 4. Its domain-based part is a strict specialization of the *c.DnSForACriminalProcess*. It has the form of a tuple (1):

```
c.DnSForAFraudulentDisbursement =
    <FraudulentDisbursement, RecognizedPlanOfAFraudulentDisbursement,
    CriminalProcessPerceivingAgent, CollectiveOfPerceivingAgent,
    FraudulentDisbursementConcept, FraudulentDisbursementEntity,
    FraudulentDisbursementInformationObject> (1)
```

The *FraudulentDisbursement* is perceived by a *CriminalProcessPerceivingAgent* who creates its description in the form of the *RecognizedPlanOfAFraudulentDisbursement* plan. This situation is constituted by *FraudulentDisbursementEntities*, for example social persons involved in the crime or documents. Also, we consider actions that are specializations of the *TaskInARecognizedPlanOfAFraudulentDisbursement*, which in turn is the specialization of the *FraudulentDisbursementConcept* (for example, we consider the important concept of *MoneyTransfer*, see section 3) that specializes *FlowOfMoneyCriminalProcessTask*. More details are given in the next subsection. The method of obtaining the ontology may be outlined as follows.

Input: *c.DnS* ontological structure, crime scenario description

Output: Ontological model of the crime with domain part aligned with *c.DnS* structure and task part expressed by rules

Method:

1. Formalize scenario in a natural-like language with variables.
2. Ask the competency questions on the basis of the resulting in step 1. description and give answers to these questions.
3. Separate ontological concepts on the basis of the answers being results in step 2.
4. Align the domain concepts obtained in step 3. with the *c.DnS* structure, creating a *c.DnS* tuple embedded into the structure (see Fig.2).
5. Formulate rules defining task-based concepts with the use of entities coming from the domain part of step 4.

3 Domain-Based Part of Fraudulent Disbursement's Ontological Model

According to the outlined method the Q1, Q2 and Q3 competency questions were asked, and further the three main groups of ontological concepts were separated (other important questions Q6-Q7 are also given).

1. Q1: What are the general entities?
The following social agents, abstracts and actions: company, single person company, institution, decision maker in company, level of document authorization, (legally valid) documents, money transfer (between companies).

2. Q2: What are the main document's flows?
The following actions: invoice flow between companies, tax statement flow.

3. Q3: What are other important flows?
The following action: goods/services flow.

4. Q4: What are the main attributes of a decision maker?
The main attribute is his/her role in the chain of command.

5. Q5: What are the entailments of a role identification?
The proof that a decision maker did particular activities resulting in a financial crime.

6. Q6: What other social agents are considered?
We consider: person not related to a company but involved in a crime, (family) relationships between such persons.

7. Q7: What are the other important data about a person?
The main data are: whether sentenced in the last 10 years, his/her criminal connections, his/her school or business connections.

8. Q8: What are the other important factors?
It is, for example, learning about averted criminal plans.

Ontology specification. The major features of the ontology are the following:

- only facts contributing to evidence or possible sanctions are kept
- answers to difficult questions are left to a human: deciding whether the work has or has not been done (requires sending an expert to the place, inspection of the construction, taking testimonies, finding that a company that presumably did the job was a straw company, having no equipment, etc.)
- relations or attributes are reduced to a minimum, for example:
 - in the first stage of the Hydra case, it is not necessary to deal with the place of construction, for the scheme would be a crime no matter where the construction was taking place (for a given jurisdiction); however, this information has to appear in the indictment
 - an invoice can be issued or received, we combine these two relations in a single one; invoices may be lost or destroyed – in cases for which these facts will be of importance, we would have to enhance the model
- knowledge about the case appears explicit as presented by facts, and implicit
 - such as regular business procedures; once the payment is approved, it is then executed and we are not interested who actually did it.

Since the previous model [10] was quite rudimentary we extended it obtaining eight groups of concepts (see the competency questions). As to the Q1 and Q6, we assume that a company has a multi-level structure of authorization. In the case of Hydra – a three-level structure: acceptance of a construction work done by a company B at a given site is first signed by a manager in a company A responsible for a work supervision at this site (*MiddleLevelManager*); this is followed by a signature of the higher level manager – a *Director* of the company responsible for supervision of all sites, who may be authorized to accept invoices and to order a payment – technically this is done by a written authorization on the back of the invoice. The *CompanysPrincipal* might not have known that the work was not being done, but he was the one who signed the contract for subcontracting and thus could be implicated. Had the *CompanysPrincipal* of A been a person who on the basis of the work acceptance document had ordered the payment of A to B, upon issuance of an invoice by B, he would be directly implicated. In reality the case was more complex entailing formalizing concepts of a complex legal document and of hierarchical chain of responsibility in a company (described by rules in section 4).

MoneyTransfer concept (Q1) codes that a money transfer has one distinctive value, it occurs at exactly one time instant between a pair of companies, and it is connected with paying for some invoice. Additionally, it is a specialization of a top level *FlowOfMoneyCriminalProcessTask* concept coming from *c.DnSForACriminalProcess* tuple. In Hydra the board consisted of 5 members, and the chairman of the board was authorized to sign documents without the consent of the others. Since no involvement of the remaining 4 members was found, here the principal is the CEO. We adopt the 3-level deep company management structure, which determines corporate lines of accountability. These 3 levels of *PersonTakingLegalDecisions*, starting from the highest levels, are: *CompanysPrincipal*, *Director* and *MiddleLevelManager*. Among concepts representing logical activities appearing in the Penal Code there is *PersonWhoFalsifiedDocument*: a (social) person connected to a company who can make decisions and is authorized to sign legally binding documents (that within a criminal activity may have the falsified content). Once falsified document can be used in further activity by a *PersonWhoUsedFalsifiedDocument* as someone who is authorized to co-sign a document and does it with the falsified one.

4 Task-Based Part of Fraudulent Disbursement's Ontological Model

This part of the ontology implements (in the form of rules) concepts concerning: a) (legal) documents and their properties, b) hierarchical chain of responsibility in a company, c) executed transactions, d) actions made by persons and e) legal sanctions. These rules can be further used to formulate queries to extract knowledge. They are built from domain-part's ontology concepts and binary relations applied to arguments being variables. The first four rules concern documents. *ContractDocument* is a document drawn up between two parties as a base of

a *Transaction* (5), signed by principals of these companies. First, the following general rule for the *ContractDocument* is defined (2):

```
Document(d), isSignedBy(d,p1), CompanysPrincipal(p1), isSignedBy(d,p2), CompanysPrincipal(p2),
differentFrom(p1,p2)
-> ContractDocument(d) (2)
```

ComplexInternalLegalDocument is a virtual hierarchical document (consisting of several physical documents), authorizing a payment (here it consists of a construction work acceptance document, and a payment authorization signature on the back of an invoice). The series of authorizations reflects the structure of the company from the lowest to the highest rank of management (3) and (4):

```
Document(d), Work(w), concerns(d,w), isSignedBy(d,p1), isSignedBy(d,p2), worksFor(p1,c),
worksFor(p2,c), hasLevelOfResponsibility(p1,l1), hasLevelOfResponsibility(p2,l2),
swrlb:lessThan(l1,l2), DifferentFrom(p1,p2), DifferentFrom(l1,l2)
-> ComplexInternalLegalDocument(d) (3)
```

```
Document(d), Work(w), concerns(d,w), isSignedBy(d,p1), isSignedOnBackOfInvoiceBy(d,p2),
worksFor(p1,c), worksFor(p2,c), hasLevelOfResponsibility(p1,l1),
hasLevelOfResponsibility(p2,l2), swrlb:lessThan(l1,l2), DifferentFrom(p1,p2),
DifferentFrom(l1,l2)
-> ComplexInternalLegalDocument(d) (4)
```

There exists also *FalsifiedComplexLegalDocument* meaning the *ComplexLegalDocument* with approval of work which was not done. The next rule concerns execution of a generalized transaction (5) involving four elements: the contract between two companies, the work, the invoice issued for work and payment.

```
ComplexInternalLegalDocument(i), ContractDocument(d), Invoice(i), Company(c1), Company(c2),
MoneyTransfer(mt), Work(w), Person(p1), Person(p2), concerns(d,w), concerns(i,w),
isSignedBy(d,p1), isSignedBy(d,p2), flowsFrom(mt,c1), flowsTo(mt,c2), isIssuedBy(i,c2),
isReceivedBy(i, c1), worksFor(p1,c1), worksFor(p2,c2), DifferentFrom(c1,c2),
DifferentFrom(p1,p2), DifferentFrom(d, i)
-> Transaction(d), hasInvoice(d, i), hasMoneyTransfer(d, mt),
transactionFrom(d, c1), transactionTo(d, c2) (5)
```

Using rules we can also ask about legal sanctions in accordance with the activities of key persons involved in the Hydra case. We take into consideration activities of three types of social agents, namely *CompanysPrincipal*, *Director* and *MiddleLevelManager*. The *MiddleLevelManager* and the *Director* (in Hydra the *Director* approved the money transfer) committed intellectual falsification. This is established beyond doubt, so they are fraudsters unconditionally. The *CompanysPrincipal* acts as unconditional fraudster only when he signed the falsified document. There are three other cases of *CompanysPrincipal*'s activity. In the first one he might have intent to commit the crime for he should have known that the work has not been done. In the second case, he might not have known that the work has not been done, so he was probably negligent. The last case deals with *CompanysPrincipal*'s conditional involvement in the crime, because additional information is needed to prove that he is the part of it. This qualification is left to an investigator or a prosecutor and may be established in several ways: a) through a guilty plea during testimonies or in court, b) with the help of other members of the scheme, testifying that he was part of it and c) by observing money transfers to his account which cannot be accounted for.

Summing up, the activities of the three types of agents form the following sequences.

1. Sequence 1
 - *MiddleLevelManager* falsifies construction work acceptance document.
 - *Director* co-signs falsified construction work acceptance document.
 - *CompanysPrincipal* accepts the document and orders payment.
2. Sequence 2
 - *MiddleLevelManager* falsifies construction work acceptance document.
 - *Director* accepts the document and orders the payment.
 - *CompanysPrincipal* should have known that the work has not been done, if he was not negligent.
3. Sequence 3
 - *MiddleLevelManager* falsifies construction work acceptance document.
 - *Director* accepts the document and orders the payment.
 - *CompanysPrincipal* might not have known that the work has not been done.
4. Sequence 4
 - *MiddleLevelManager* falsifies construction work acceptance document.
 - *Director* accepts the document and orders the payment.
 - *CompanysPrincipal* may be a part of the crime scheme, what is left to be proved by an investigator or a prosecutor.

On the basis of these four activity sequences, rules were defined to obtain results from the current state of the knowledge base, enabling to determine which sequence of activities is appropriate. For example, (6) was defined for activities in *FraudulentDisbursementCrime*, according to the sequence 1 and the rule (7) to express being in complicity:

```
FalsifiedComplexInternalLegalDocument(d), Company(c),MoneyTransfer(m),CompanysPrincipal(p3),
Director(p2), MiddleLevelManager(p1), accepts(p3,d), orders(p3,m), isSignedBy(d,p1),
isSignedOnBackOfInvoiceBy(d,p2), knowsAbout(p3,d),worksFor(p1,c), worksFor(p2,c),
worksFor(p3,c), DifferentFrom(p1,p2), DifferentFrom(p1,p3),DifferentFrom(p2, p3)
-> FraudulentDisbursementCrime(p1, "1"), FraudulentDisbursementCrime(p2, "1"),
FraudulentDisbursementCrime(p3, "1") (6)
```

```
Company(c), CompanysPrincipal(p3), Director(p2), MiddleLevelManager(p1), worksFor(p1,c),
worksFor(p2,c), worksFor(p3,c), FraudulentDisbursementCrime(p1,"1"),
FraudulentDisbursementCrime(p2,"1"), FraudulentDisbursementCrime(p3,"1"),
DifferentFrom(p1,p2), DifferentFrom(p1,p3), DifferentFrom(p2, p3)
-> inComplicityWith(p1, p2), inComplicityWith(p1, p3), inComplicityWith(p2, p3) (7)
```

At the end we give exemplary rules defining legal sanctions reached by an expert prosecutor and the judge assessing previously described activities. Considering the fraudulent disbursement sanctions in the Polish Penal Code (PC) it appears that most defendants were accused as follows:

1. Art. 296 §1-3 PC – strictly: negligence, leading to damage to a company (for personal benefit); also, fraudulent disbursement.
2. Art. 296 §4 PC – unknowing negligence leading to damage to a company.
3. Art. 284 §2 PC – personal benefit resulting from activities sanctioned by art. 296.
4. Art. 294 §1 PC – as in 284 §2 PC with regard to property of considerable value.
5. Art. 286 §1 PC – fraud (intentionally deceiving a person: here a legal person, which results in a damage to the company).
6. Art. 271 §3 PC – lying or issuance (signing) a false document with regard to a circumstance having legal significance.
7. Art. 273 §1 PC – using a document mentioned in art. 271 §3 PC.
8. Art. 299 §1 and §5 PC – money laundering (conscious and with other persons).
9. Art. 18 §1 PC – directing illegal activity performed by another person.

It appeared that there exist three types of accusation in the Hydra case. All of them are expressed by separate rules (8), (9) and (10).

```
Art_271_1(a5),Art_273(a4),Art_284_2(a3), Art_286_1(a1), Art_294_1(a2),
FalsifiedComplexInternalLegalDocument(d), NotInConspiracy(p3),CompanysPrincipal(p3),
Director(p2), MiddleLevelManager(p1),isSignedBy(d, p1),inComplicityWith(p1, p2),
DifferentFrom(a1, a2),DifferentFrom(a1, a3),DifferentFrom(a1, a4),
DifferentFrom(a1, a5),DifferentFrom(a2, a3),DifferentFrom(a2, a4),
DifferentFrom(a2, a5),DifferentFrom(a3, a4),DifferentFrom(a3, a5),
DifferentFrom(a4, a5),DifferentFrom(p1, p2),DifferentFrom(p1, p3),
DifferentFrom(p2, p3)
-> Innocent(p3), fallsUnder(p1, a2), fallsUnder(p1, a5), fallsUnder(p2, a1),
fallsUnder(p2, a2) (8)
```

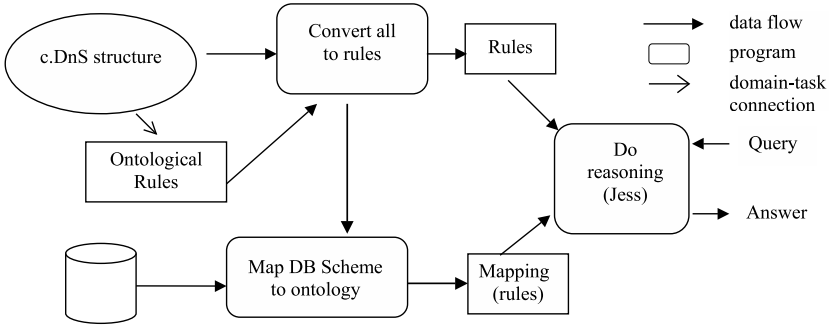


Fig. 3. Database querying within SDL (Semantic Data Library) [11]

```
FalsifiedComplexInternalLegalDocument(d), Art_271_1(a5), Art_273(a4), Art_284_2(a3),
Art_286_1(a1), Art_294_1(a2), Art_296_4(a6), Negligent(p3), NotInConspiracy(p3),
CompansPrincipal(p3), Director(p2), MiddleLevelManager(p1), isSignedBy(d,p1),
inComplicityWith(p1,p2),DifferentFrom(p1, p2), DifferentFrom(p1,p3), DifferentFrom(p2,p3)
-> fallsUnder(p1,a2),fallsUnder(p1,a5),fallsUnder(p2,a1), fallsUnder(p2,a2),
fallsUnder(p2,a3),fallsUnder(p2,a4), fallsUnder(p3,a6) (9)
```

```
FalsifiedComplexInternalLegalDocument(d), Art_271_1(a5), Art_273(a4),Art_284_2(a3),
Art_294_1(a2), Art_296(a1), Organizer(p3), CompansPrincipal(p3), Director(p2),
MiddleLevelManager(p1),isSignedBy(d,p1),inComplicityWith(p1,p2), inComplicityWith(p1,p3),
nComplicityWith(p2,p3),DifferentFrom(p1,p2), DifferentFrom(p1,p3), DifferentFrom(p2,p3)
-> fallsUnder(p1,a1),fallsUnder(p1,a2),fallsUnder(p1,a3), fallsUnder(p1,a5),
fallsUnder(p2,a1),fallsUnder(p2,a2),fallsUnder(p2,a3),fallsUnder(p2,a4),
fallsUnder(p3,a1), fallsUnder(p3, a2), fallsUnder(p3, a3) (10)
```

We have verified our ontology with the Pellet-2.0.1 reasoner. We did some experiments with database querying with the use of ours SDL (Semantic Data Library), the ontology and the Jess engine (Java Expert System Shell, <http://jessrules.com>). Unfortunately, due to the lack of space the details of experiments are not shown in this paper – they are reported in [11], [2]. Here, we give a scheme of this activity, depicted in Fig. 3, where *c.DnS structure* and *Ontological Rules* are those of Fig.2.

5 Conclusions

The presented model consists of a domain description and a task specification in the form of rules (see Fig. 2). Our analysis accounts only for crimes of people associated with Hydra – company A (Fig.1). Here, we do not present rules for sanctions for people in other involved companies nor give sanctions for money laundering. The model is not yet able to determine the duration of an appropriate penalty. In the future, we intend to demonstrate that we can describe not just one case but a broad class of economic crimes, such as: causing asset misappropriation and causing damage to a company. After performing quite a number of reasoning experiments on the five most common economic crime mechanisms in Poland ([11], [12]), we are convinced that a general model can be constructed that handles a few most common economic crimes with 85% use of all pertinent facts (the Hydra case is somewhat easier than average). To our knowledge, the work on mapping of

crime activities into criminal law articles has been done [3] only for cyber crimes, which have a much narrower scope, although using result of work [15] it could be straightforward for the case. Currently, we are working on implementation of the knowledge base and generation of artificial data to test our approach. We present here the part of our ontology concerning the fraudulent disbursement. It is relatively small – the whole economic crime’s ontology will be enhanced once we add other fraud type typologies (for example, fuel crime, see Fig. 2 and [5]).

References

1. Bak, J., Jędrzejek, C., Falkowski, M.: Usage of the Jess Engine, Rules and Ontology to Query a Relational Database. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 216–230. Springer, Heidelberg (2009)
2. Bak, J., Jędrzejek, C., Falkowski, M.: Application of an Ontology-Based and Rule-Based Model to Selected Economic Crimes: Fraudulent Disbursement and Money Laundering. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 210–224. Springer, Heidelberg (2010)
3. Bezzazi, H.: Building an ontology that helps identify articles that apply to a cybercrime case. In: Proc. of ICSOFT 2007, Barcelona, Spain, pp. 179–185 (2007)
4. Cybulka, J.: Applying the c.DnS Design Pattern to Obtain an Ontology for Investigation Management System. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 516–527. Springer, Heidelberg (2009)
5. Cybulka, J.: Fuel Crime Conceptualization through Specialization of Ontology for Investigation Management System. In: Nguyen, N.T., Kowalczyk, R. (eds.) Transactions on CCI II. LNCS, vol. 6450, pp. 123–146. Springer, Heidelberg (2010)
6. Gangemi, A.: The c.DnS ontology,
<http://www.loa-cnr.it/ontologies/collint.owl>
7. Gangemi, A., Lehmann, J., Catenacci, C.: Norms and plans as unification criteria for social collectives. In: Proc. of Dagstuhl Sem. 07122, vol. II, pp. 48–87 (2007); ISSN 1862-4405
8. Gangemi, A., Lehmann, J., Presutti, V., Nissim, M., Catenacci, C.: C-ODO: an OWL Meta-model for Collaborative Ontology Design. In: Proc. of WWW 2007 Workshop on CKC 2007, Banff, Canada (2007)
9. Gómez-Pérez, A., Suárez de Figueroa Baonza, M., C., Villazón, B.: Neon Methodology for Building Ontology Networks: Ontology Specification, NeOn Deliv. D5.4.1(2008), <http://www.neon-project.org>
10. Jędrzejek, C., Cybulka, J., Bąk, J.: Application Ontology-based Crime Model for a Selected Economy Crime. In: Proc. of CMS, Kraków, Poland, pp. 71–74 (2009)
11. Jędrzejek, C., Falkowski, M., Smoleński, M.: Link Analysis of Fuel Laundering Scams and Implications of Results for Scheme Understanding and Prosecutor Strategy. In: Proc. of JURIX 2009, Rotterdam, The Netherlands (2009)
12. Martinek, J.: Hydra case formal structural description. Tech. rep., PPBW, Poznań University of Technology, Poznań, Poland (2008)
13. Parton, T., Rajarao, V., Skalak, S.: Pricewaterhouse Coopers’ Global Economic Crime Survey (2009), <http://www.pwc.com/gx/en/economic-crime-survey/download-conomic-crime-people-culture-controls.html>
14. Więckowski, J.: Hydra case indictment analysis, District Public Prosecutor in Katowice, Poland, personal communication (2009)
15. Wyner, A.: An ontology in OWL for legal case-based reasoning. Artificial Intelligence and Law 3, 361–387 (2008)

A New Mechanism for Negotiations in Multi-Agent Systems Based on ARTMAP Artificial Neural Network*

R. Beheshti and N. Mozayani

Department of computer engineering, Iran university of science and technology,
Narmak, Tehran, Iran

beheshti@comp.iust.ac.ir, mozayani@iust.ac.ir

Abstract. Any rational agent involving in a multi-agent systems negotiation tries to optimize the negotiation outcome based on its interests or utility function. Negotiations in multi-agent systems are usually complex, and a lot of variables exist which affect the agents' decisions. This becomes more visible in competitive or multi-issue types of negotiations. So, the negotiator agents need an efficient mechanism to do well. The key solution to this type of problems is employing a powerful and operative learning method. An agent tries to learn information it obtains from its environment in order to make the best decisions during the negotiations. In real-world multi-agent negotiations, the main source of usable data is the negotiators' behaviors. So, a good learning approach should be able to extract the buried information in the 'negotiation history'. In this work, we used an ARTMAP artificial neural network as a powerful and efficient learning tool. The main role of this component is to predict other agents' actions/offers in the next rounds of negotiation. When an agent finds out what are the most possible offers which will be proposed, it can predict the outcomes of its decisions. In addition, a new method to apply this information and determine next moves in a negotiation is proposed. The obtained experimental results show that this method can be used effectively in real multi-agent negotiations.

Keywords: Multi-agent systems, Negotiation, Learning, ARTMAP artificial neural network.

1 Introduction and Background

Negotiations in multi-agent systems (MAS) are one the challenging domains that still need more research and study in order to approach the efficiency and stability of negotiations between human beings. These negotiations become more complex when they involve multi-lateral problems and include many rational agents with different or even conflicting desires. Agents usually try to maximize their utility and minimize the negotiation duration, especially in non-cooperative domains. All of these make importance of a learning mechanism more obvious.

* This research was supported by Iran Research Institute for ICT (ITRC).

Agents use a learning tool to find out how to proceed the negotiation. The agents involving in different negotiations usually show a similar behavior and attitude, and some kind of pattern can usually be find in their activities. In real MAS negotiations this behavior of agents is the main usable source of information which can be used in order to learn agents' behavior and predict their next moves. MAS negotiations usually take place within sequential rounds that each agent proposes its offer/s in every round. That is why most of the existing works on MAS negotiations are based on learning methods which use the negotiation history or log [5]. Artificial neural networks (ANNs) are one of most common learning tools which the existing negotiation models already used.

In [3] an approach to modeling the negotiation process in a time-series fashion using artificial neural network is proposed. In essence, the network uses information about past offers and the current proposed offer to simulate expected counteroffers. On the basis of the model's prediction, what-if analysis of counteroffers can be done with the purpose of optimizing the current offer. The neural network has been trained using the Levenberg-Marquardt algorithm with Bayesian Regularization. AN RBF neural network technology in multi-agent negotiations is introduced in [8]. This method establishes a bilateral-multi-issue negotiation model, and defines corresponding negotiation algorithm and utility evaluation functions. Negotiation agents learn to change their belief of the environment and other agents, using RBF neural network, thus to determine the inference strategy in negotiation. An artificial neural network-based predictive model with application for forecasting the supplier's bid prices in supplier selection negotiation process (SSNP) is developed in [7]. By means of this model, demander can foresee the relationship between its alternative bids and corresponding supplier's next bid prices in advance. The purpose of this work is applying the model's forecast ability to provide negotiation supports or recommendations for demander in deciding the better current bid price to decrease meaningless negotiation times, reduce procurement cost, improve negotiation efficiency or shorten supplier selection lead-time in SSNP. In [9] an adaptive feed-forward ANN is used as a learning capability to model other agent negotiation strategies. Another learning based method which employs ANN is produced in [12]. The aim of this method is to implement interactions between agents and guarantees the profits of the participants for reciprocity. In the system, each agent has a learning capability implemented by an artificial neural network to generate sequential offers and can be trained by the previous offers that have been rejected by the other agent. With this negotiation model, agents can negotiate with each other over a set of different issues of a product on behalf of the real-world parties they represent. In [11], a data ratios method is proposed as the input of the neural network technique to explore the learning in automated negotiation with the negotiation decision functions (NDFs) developed in [6]. The concession tactic and weight of every issue offered by the opponent are learned from this process exactly. After learning, a trade-off mechanism is applied to achieve better negotiation result on the distance to Pareto optimal solution. An ANN and GA-assistant method which uses genetic algorithm to predict the behavior of opponents and employs MLP and RBF ANNs to refine the results is proposed in [10].

In this paper, we will introduce our method which employs ARTMAP ANN as a learning tool like the described methods. We believe that this type of ANN is the most fitting kind of ANNs that can be efficiently used in multi-agent negotiations. Firstly, in section 2, ARTMAP ANNs and their main favorable characteristics will be briefly discussed. In section 3 the proposed method will be illustrated. Section 4 discusses the experimental results. Finally, section 5 includes concluding remarks.

2 Using Neural Networks in Multi-Agent Negotiations

The learning mechanism that is used in MAS negotiations domains should be simple and powerful. This usually makes a trade-off between simplicity and learning ability. For example, although evolutionary based learning methods have satisfying learning ability, but they generally need considerable computing time. ANNs have both characteristics. Artificial Neural networks are among most powerful learning tools. They efficiently can be used in real-time environments. After training phase -which may last long, they can generate desired results immediately. Artificial neural networks like other pattern recognition tools, may be used in supervised or unsupervised learning based applications. In supervised domains, class type of each dataset entry is known, and a neural network tries to adapt itself to the available data. In unsupervised cases, the class types are not known.

As mentioned earlier, it is assumed that every agent involved in a negotiation has a history or archive of others' previous actions. Negotiation mechanisms usually try to maximize payoff or utility of an agent after the negotiation ends. Experiments show that agents usually repeat their behavior and use similar strategies in the same situations and show same eagerness to concede [2]. So behavior history of an agent can be used as a useful source for predicting its behavior in the future. From this point, we reach to the idea of employing Neural Networks as a prediction tool. For this purpose, as we have a kind of training set (transactions history) it seems that supervised learning types of Neural Networks are more appropriate than unsupervised ones. In addition, the agents which involve in the negotiations, do not have complete knowledge about other agents. Like the negotiations that take place in human societies, they can usually observe only what other agents offer in each round. Any additional assumption about available data may limit the generality of a negotiation mechanism. When a negotiation starts, a negotiator agent records the offers that other agents propose. As the negotiation proceeds, these records increase, and the agent's data grows. So, the artificial neural network which is used here should be able to adjust itself to this new added data. This means that ANN must adjust itself to the new data efficiently and with minimum changes in the network. In ANNs' term, the ANN should be able to learn incrementally.

When these two main characteristics are put next to each other (effective supervised and incremental learning ability), we reach to the ARTMAP ANN. This type of ANN employs supervised learning and is able to adapt itself with the

Table 1. Supervised & Incremental learning support of main types of ANNs

	MLP	Recurrent Nets.	RBF	Kohonen	Art
Supervised	✓	✓	✓	×	✓
Incremental	×	×	×	✓	✓

new entering data. In ARTMAP, adding new samples does not need to retrain all of the network with the enlarged training set until a new stable state is reached. Number of class nodes is not pre-determined and fixed. The fixed number of classes – which is a common property of ANNs – may result to under or over classification. In this case, there is no way to add a new class node (unless a free class node happens to be close to the new input). Any new input X has to be classified into one of existing classes (causing one to win), no matter how far away X is from the winner, and generally, no control of the degree of similarity exists. Table 1 shows several main types of ANNs. As the table shows, other type of ANNs (especially their classic versions) do not have both supervised and incremental learning ability. This makes ARTMAP one of the best candidates to be used in MAS negotiations.

The main problem of general ANNs is that gradually decreasing the gain parameter, usually freezes the network’s weights, and consequently no flexible reaction to new data is made. ARTMAP, that is based on the theory of ‘ART’, could solve these issues efficiently.

Adaptive Resonance Theory (ART) is a theory developed by S. Grossberg and G. Carpenter on aspects of how the brain processes information [1]. It describes a number of Neural Network models which use supervised and unsupervised learning methods. This paper focuses on the prediction ability of ART Neural Networks. This ability is applied by negotiator agents to predict other agents’ next offer/s. For more details about ARTMAP the reader is referred to [4].

3 The Proposed Method

In describing the proposed method, we firstly introduce the configuration of applied ANN. Fig. 1 shows the general structure of applied ANN. The existing idea here is that the ANN receives last offers which the agents proposed in the previous round of negotiation and then predicts next offer which each agent will propose. The inputs of the network are all of offers proposed by the negotiators in the previous round of negotiation plus the number of current round. The outputs of the network are the predicted offers of other negotiator agents in the next round. In this manner, in a negotiation containing ‘ n ’ agents, the number of the network’s inputs is $n+1$, and the number of outputs is $n-1$ (the number of other agents). The history of previous negotiations is the main source of information in real negotiation domains. Like the human negotiations, the previous offers of other negotiators and the round in which the negotiation is being performed are the most important factors that determine the offer the negotiator propose. In

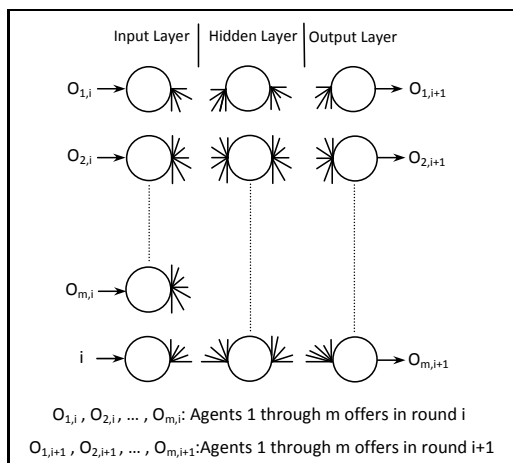


Fig. 1. Simple schema of the proposed ARTMAP ANN

other words, in a same round of two negotiations which the previous offers are the same, an agent will probably propose the same offer. Our experiments show that this probability is high. As the negotiations proceed, the ANN learns more, and produces predictions that are more accurate.

Before proposing its offer in a round, a negotiator agent employs the ANN to find out the probable offers of other agents. As we discussed, “payoff” and “time” are two main factors that agents want to optimize. So, an agent uses the information that it obtains from the ANN to reach these goals.

Time effect is taken into account in MAS negotiations through different approaches. Here, we use a method which often is used in bargaining domains. In these domains a discount ratio, ‘ Ω ’, is used to represent the effect of time. This value determines the value of an offer in the next rounds, knowing its value in current round. Higher Ω value implies less importance of time for an agent. For example, if Ω is equal to 0.9 and the utility value of an arbitrary option like o_1 in the round i is 100, the utility value of this option in round $i + 1$ will be 90.

Here, we illustrate our scenario for the agents to use the ANN results in order to choose their next offer. In each round, agents propose their offers. Other characteristics of negotiation domains like being single-issue or multi-issue, bilateral or multi-lateral, competitive or cooperative do not have a direct effect on our method, and the proposed method can be efficiently used in all of these domains. Every negotiation has a set of valid or admissible offers that negotiators can propose. As described in the previous paragraph an agent’s first concern is to maximize its utility and after this it tries to minimize the negotiation time (because in all of MAS domains passing time is costly). Like any rational and self-interested agent, in each round the negotiator agent firstly determines the offer which maximizes its utility. We call this, ‘maximal potion’. This offer usually is selected by the agent among a set of offers that the agent can propose. This may be a simple sorting process or may require complex computations.


```

1- Generate a random number (R) between 0 and 1
2- Sort the available options, and determine the option with maximum utility value \\maximal option
3- IF (R < Exploitation Probability) THEN
    \\The ARTMAP ANN is trained and its results including the agents' next offer are available
4- Sort the options which guarantee agreement in the current round [based on the learning results], and
   determine the option having maximal utility \\rational option
5- IF (rational option's utility > maximal option utility AND rational option's utility > agent's minimum
   utility) THEN
6- Propose the rational option
7- ELSE IF (Maximal option's utility > agent's minimum utility) THEN
8- Propose the maximal option
9- ELSE
10- Quite the negotiation
11- ELSE
12- Propose the maximal option \\learning is not used

```

Fig. 2. Pseudo-code of the proposed method

After this, the agent should take into account the ‘time’ measure, too. In this step, the agent checks the results of the ANN. Considering the previous offers which the agents proposed in earlier rounds in addition to the predicted offers, if the maximal option guarantees reaching agreement (and consequently, ending the negotiation) the agent will propose the maximal offer. If the maximal option does not ensure reaching agreement, the agent will check other options it can propose. If there was an option that guarantees agreement, and its utility value is greater than the utility of maximal option in the next round, the agent would propose this option. We call this option “Rational Option”. Rational option is the maximal option between the options ensure reaching agreement in the current round. If proposing the maximal option ends the negotiation in the current round with an agreement, the maximal and rational option will be the same. The idea here is that the maximal option –in the best case– will be applicable in the next round (not in the current round), but the rational option although has less utility than the maximal option, but in practice the agent would gain more utility by offering the rational option. In other words, in this case the rational option becomes maximal option. Fig. 2 shows pseudo-code of the proposed method for an arbitrary agent.

To illustrate this idea, consider this example. Imagine an agent like a_1 that can choose its offer from a set of three options: o_1, o_5, o_6 in an arbitrary round, ‘ i ’. It is assumed in this example that each agent may offer only one option in each round, and these options should be chosen from an option pool containing six options. The agreement is met when a permutation of proposed options can be found that all of six options are assigned to the agents based on their offers. The option which maximizes the agent’s utility function (maximal option), is o_1 with the utility of 100. o_6 is the rational option which guarantees agreement, and its utility is greater than o_1 ’s in the next round. Fig. 3 shows this scenario.

Some minute details of the proposed method are still unclear which will be illustrated here. When the negotiations starts, agents’ history of previous proposed offers are empty, and consequently there is no record for the ANN to

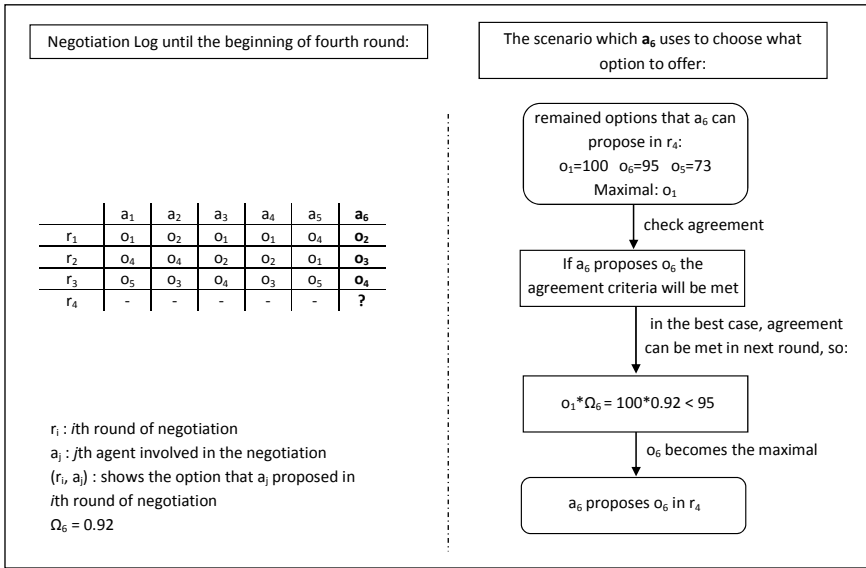


Fig. 3. Choosing the next offer using information obtained after learning

learn agents’ behavior. To resolve this problem, a “exploitation probability” is assigned to each agent which indicate the probability that the agent employs the ANN’s results to choose its next offer. This probability increases as the negotiation proceeds. Another issue is that each agent have a minimum utility value that indicates until when the agent is going to participant in the negotiation. An agent decide to leave the negotiation table, when it finds out that it cannot gain its minimum utility value by continuing the current negotiation.

4 Experimental Results

In this section, we illustrate several experiments which were accomplished in order to examine the proposed method. The negotiations are about a task allocation problem. There are n different tasks and n agents who want to choose the task having maximum utility. The agreement is reached when a permutation of offered options can be found that each task is assigned to one agent. Assignment of a task to an agent is possible only when that option is previously proposed by that agent. In the following experiments, the number of negotiator agents (which is equal to the number of tasks) are 20 and 100. These agents participate in 20 sequential negotiations. All of these negotiations are about the task assigning problem, but each time 20(100) tasks are randomly chosen from a pool of tasks including 40(200) different tasks. The configurations of negotiations are the same for both of methods being studied. This means that the tasks existing in each of 20 sequential negotiations are used to evaluate both of methods. Table 2 shows the configurations of the ARTMAP ANN which is used in the experiments. Number of inputs –as was shown in Fig. 1– is equal to number of

Table 2. The configurations of ARTMAP ANN used in the experiments

Inputs Num.	Outputs Num.	Net's weights	Vigilance Param.	Epochs Num.
21/101	20/100	all initially equal to 1	0.75	100

negotiator agents plus the number of round in which those offers were proposed. Number of outputs is equal to the number of involved agents. In this manner, each record of an agent’s history – which is used during the training phase of the ANN– includes $2n + 1$ fields: n fields for all of agents’ offers in a negotiation’s round, n fields for their offer in the next round, and the last remaining field for the round number. The initial values of the ANN’s weights are set to 1, and each ANN is trained within 100 epochs.

The proposed method is compared with another method which is introduced in [8]. In this work, an RBF ANN is used by the agents in order to determine their proposals in each round. The RBF ANN’s settings are like the ones which are described in the original work. The applied radial basic function in the network is illustrated there, too. The inputs and outputs are the same as above.

“Negotiation time” and “Agents’ utility” are two main factors that each agent wants to optimize them. Accordingly, these two measures are used in order to examine two available methods. The time (duration) of negotiation shows the number of rounds that it takes until the negotiation ends. The negotiation ends when the agreement is reached. Here, agreement is guaranteed, because the negotiators cannot propose repeated offers. Accordingly, it takes at most n rounds in order to reach agreement (after n round a permutation exists). Furthermore, the agents’ utility refers to the average of the utility values that all of negotiator agents gain after the negotiation ends. A better negotiation mechanism should be able to decrease the duration of negotiation while maximizing the agents’ utility. Each task has a different utility value from each agent’s point of view. These utility values are randomly produced between 0 and 100.

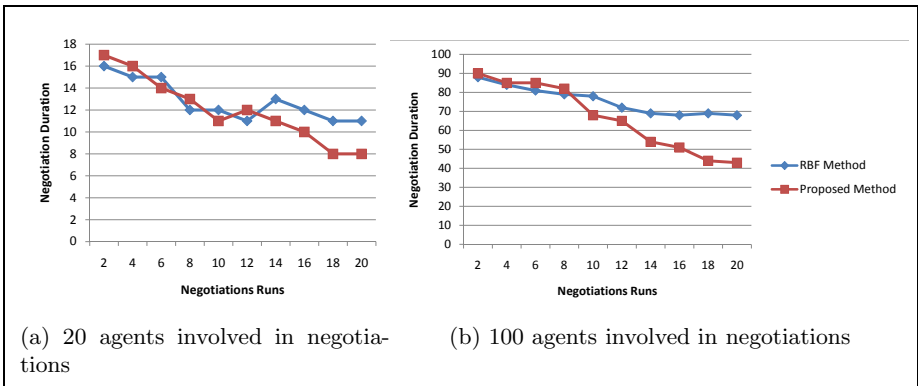


Fig. 4. Negotiation time in 20 sequential negotiations

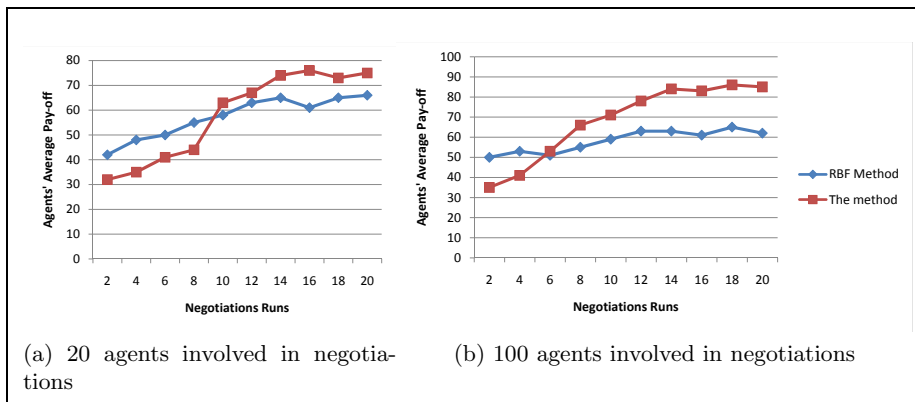


Fig. 5. Average agents' pay-off, in 20 sequential negotiations

Fig. 4a shows the duration of negotiations between 20 agents. When the number of negotiations increases, both of methods were able to learn almost well. The proposed method's results surpass the other method after the thirteenth round. In the Fig. 4b negotiation time between 100 agents is displayed. In these experiments our method totally outperforms the RBF based method, where the second method's results approximately stays fixed after the 13th round.

Agents' pay-off(utility) is the second measure which is studied in Fig. 5a and Fig. 5b. In these figures the average pay-off of all of agents involving in the negotiations are shown in different rounds. As the number of agents increases (from 20 to 100) our method's shows better performance. This means that the proposed method can be efficiently used in domains with many agents which makes the negotiation domain more complex.

5 Conclusion

In this work a new mechanism for negotiations in multi-agent systems is proposed. This methods relies on ARTMAP artificial neural network to learn. A negotiator agent employs this learning tool in order to determine the most possible offers which other agents will propose. Then, the agent checks the possibility of each option to end the negotiation with an agreement. Based on the utility value of each option, the agent chooses its next offer. The described characteristics of ARTMAP in the paper shows that this type of ANN which is embedded within our method is one the best candidates for our usage. Using ARTMAP allows us to implement the method in real-time applications. The proposed method was examined within several experiments and has obtained satisfactory results in its first applications.

References

1. Arbib, M.: The handbook of brain theory and neural networks. MIT Press, Cambridge (2002)
2. Bui, H., Kieronska, D., Venkatesh, S.: Learning other agents' preferences in multi-agent negotiation. In: Proceedings of the National Conference on Artificial Intelligence, pp. 114–119 (1996)
3. Carbonneau, R., Kersten, G., Vahidov, R.: Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications* 34(2), 1266–1273 (2008)
4. Carpenter, G., Grossberg, S., Reynolds, J.: ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural networks* 4(5), 565–588 (1991)
5. Esmaili, A., Mozayani, N.: Improving Multi-agent Negotiations Using Multi-Objective PSO Algorithm. In: *Agent and Multi-Agent Systems: Technologies and Applications*, pp. 92–101 (2010)
6. Faratin, P., Sierra, C., Jennings, N.: Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems* 24(3-4), 159–182 (1998)
7. Lee, C., Ou-Yang, C.: A neural networks approach for forecasting the supplier's bid prices in supplier selection negotiation process. *Expert Systems with Applications* 36(2), 2961–2970 (2009)
8. Liu, N., Zheng, D., Xiong, Y.: Multi-agent Negotiation Model Based on RBF Neural Network Learning Mechanism. In: *International Symposium on Intelligent Information Technology Application Workshops, IITAW 2008*, pp. 133–136 (2008)
9. Oprea, M.: An adaptive negotiation model for agent-based electronic commerce. *Studies in Informatics and Control* 11(3), 271–279 (2002)
10. Papaioannou, I., Roussaki, I., Anagnostou, M.: Neural networks against genetic algorithms for negotiating agent behaviour prediction. *Web Intelligence and Agent Systems* 6(2), 217–233 (2008)
11. Rau, H., Tsai, M.H., Chen, C.W., Shiang, W.J.: Learning-based automated negotiation between shipper and forwarder. *Comput. Ind. Eng.* 51(3), 464–481 (2006), <http://dx.doi.org/10.1016/j.cie.2006.08.008>
12. Zeng, Z.M., Meng, B., Zeng, Y.Y.: An adaptive learning method in automated negotiation based on artificial neural network. In: *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 383–387 (2005)

A Multi-Agent Architectural Pattern for Wrapping Off-the-Shelf Components

Sodany Kiv¹, Yves Wautelet², and Manuel Kolp¹

¹ Université catholique de Louvain, Belgium
{sodany.kiv, manuel.kolp}@uclouvain.be

² Hogeschool-Universiteit Brussel, Belgium
yves.wautelet@hubrussel.be

Abstract. The benefits of off-the-shelf component-based software development are essentially lower cost and shorter development time. Nevertheless, the integration of components is seldom trivial. Indeed, developers can face black boxes or products that are poorly documented and specified. Mismatches between components functionality and system requirements are then difficult to address. To deal with such issues, this paper proposes to use an architectural pattern for wrapping software components. To that extend, we develop a multi-agent system architecture – the wrapper – that offers an abstraction layer over the components. The architecture is in charge of the functional decomposition, coordination and delegation of the client requests to the components at disposal.

1 Introduction

Component-based system development (CBSD) implements the idea of building new systems by selecting appropriate components and assembling them within a well-defined software architecture [3]. Although such an approach promises significant benefits, there are some technical problems that limit their use [13,16]. This paper proposes to resolve some of those issues by defining a *wrapping abstract layer* in the form of a Multi-Agent System (MAS) architecture. The essence of the technique is to use a “wrapper” program (materialized through source code) between the components layer and their environment to deal with the flows of control and data going to and/or coming from a wrapped component. This means that a commercial off-the-shelf (COTS) product is embedded in and controlled by code that is not developed by its vendor.

Such an approach is not genuine; besides industrial frameworks that do not integrate intelligent technology, several MAS architectures have been proposed in the literature (see for example [6]). Our approach goes further by providing an intelligent coordination mechanism as well as architectural configurability. In other words, the architectural pattern proposed in this paper is generic but can be adapted to a particular business logic issued of a project-specific domain model. Indeed, this contribution is intended to be encapsulated in the requirements-driven CBSD methodology described in [17] which covers the whole (iterative) software development life cycle. [17] nevertheless lacks of a proper architectural description and this paper is intended to fill this gap.

2 Related Work

The primary goal of CBSD is to build from many COTS products a system that meets the functional and non-functional requirements defined by the stakeholders. [14][15] emphasize the importance of the system architecture and the use of agent technology in such a context and the issues addressed by wrapping COTS components are depicted in [16]. Some CBSD methods, such as OTSO (Off-The-Shelf Option) [10], the STACE (Social-Technical Approach to COTS Evaluation) Framework [11], and the PORE (Procurement-Oriented Requirements Engineering) Method [12] emphasize the importance of requirements analysis in order to conduct a successful selection that satisfies the customer. In that perspective, [17] defines an iterative CBSD methodology. This method is requirements-driven and can be linked (or enriched with) the architectural pattern proposed here on the basis of the capability concept. Indeed, capabilities are first defined in [2] as *the ability of an actor of defining, choosing and executing a plan for the fulfillment of a goal, given certain world conditions and in presence of a specific event* and, [17], proposes a formalized monitoring process of the MAS evolution at runtime through capabilities. Since our MAS architecture is driven by this similar concept (see Section 3.2) it can be said to be compliant with the methodology proposed in [17], be encapsulated at its design stage to furnish a configurable basis for components' integration.

Finally, the software industry has already adopted CBSD frameworks like *J2EE* [1], *OMG CORBA* [7] or *Microsoft Component Object Model-based technologies (DCOM)* [5]. Those object-oriented frameworks provide technologies to integrate software components in enterprise developments; their main drawbacks are the absence of dynamic allocation during the functional execution and their exclusive technology dependence. The use of agent technology in such a context allows using an intelligent coordination mechanism together with project-specific configurability. Indeed, [6] also propose a wrapping MAS architecture but their proposal is not configurable and only overviewed from a design perspective. Moreover, their MAS also remains limited with poor coordination mechanisms.

3 The MAS Architecture

This section describes the MAS architecture within its social, intentional, communicational and dynamic dimension as defined by the SKwyRL framework in [9]. For this purpose, we will make use of i^* (i-star) [18] and Agent-UML (AUML) [8] models. This architecture constitutes the wrapping abstraction layer.

3.1 Social Dimension

The social dimension focuses on the identification of relevant agents in the system and their goal interdependencies; those are represented graphically using i^* strategic dependency models.

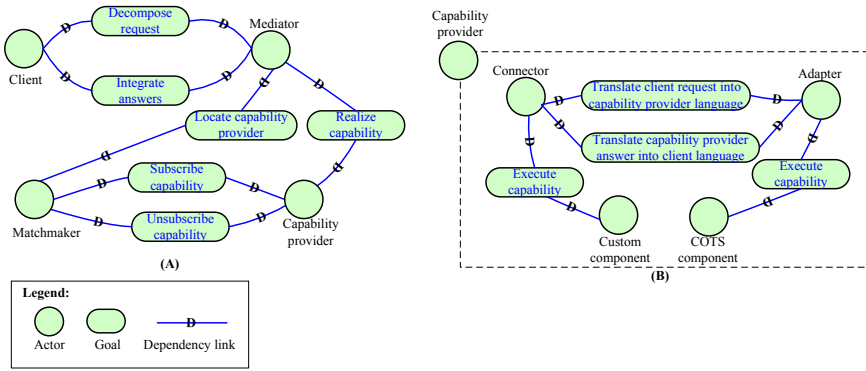


Fig. 1. Social dimension. High level view (a) and capability provider rationale (b).

High Level View: MAS Behaviour. Figure 1 (a) illustrates the social dimension of the proposed MAS through an high level view of its architecture. There are four types of agents: **Client**, **Mediator**, **Matchmaker** and **Capability Provider**. The **Client** agent is designated to a user interface to interact with the user including receive user request and display result. When receiving the client request, the **Client** agent forwards the request to the **Mediator** agent. The **Mediator** agent will decompose the request into sub-requests and ask the **Matchmaker** agent to find the corresponding capability providers. The **Mediator** agent will obtain necessary information, such as the I/O type, pre-condition and post-condition, and the location of the remote host, for remote execution. The **Mediator** agent will communicate with the capability providers in order to get answer of each sub-request and integrate these answers to formulate final result and send it to the **Client** agent.

Low Level View: Capability Provider Rationale. Figure 1 (b) focuses on depicting the **Capability Provider** issued of the connection with a particular component. It is composed of several sub-agents playing different roles. The **Capability Provider** depends on the **Connector** agent to interact with its environment. It is responsible for advertising the capabilities to the **Matchmaker** agent, receiving requests from outside and invoking corresponding capabilities. If the capability to be invoked is offered by the component, the **Adapter** agent may be used to translate the client request into the language of the component and the answer from the latter into the client language. Otherwise, the **Connector** agent can directly invoke the corresponding custom capability.

3.2 Intentional Dimension

The intentional dimension aims at modeling agent reasoning. In this dimension, we identify the capabilities provided by each agent that can be used to achieve the goal dependencies. Definition and formalizations of the capability concept are given in [217].

Table 1. Capabilities of our MAS architecture at the high level view

Capability Name	Informal Definition	Agent
SendRequest	Send a Request to Mediator	Client
AnalyseClientRequest	Analyse the client request	Mediator
DecomposeClientRequest	Decompose the client requests into sub-requests	Mediator
SendCapability-InformationRequest	Send a capability information request to Mediator	Mediator
QueryAvailability	Query the knowledge base to find the subscribe capability provider that offers the capability requested by Mediator	Matchmaker
SendCapability-Information	Send information of the capability requested to Mediator	Matchmaker
HandleMMRefusal	Handle a negative answer from Matchmaker	Mediator
RecordCapability-Information	Record information of the capability requested	Mediator
SendSubRequest	Send sub-request to the capability provider	Mediator
RecordSubResult	Record sub-results provided from capability provider	Mediator
HandleSubRequestFailure	Handle the negative answer from capability provider	Mediator
IntegrateSubResult	Integrate sub-results provided from capability providers	Mediator
SendIntegratedResult	Send integrated result to client	Mediator
RecordResult	Record the answer sent by Mediator	Client
HandleSPFailure	Handle the failure from capability provider cannot provide the answer for client request	Mediator
HandleFailure	Handle the failure when Mediator cannot provide the answer for client request	Client

Table 1 lists the capabilities of our MAS architecture, their informal definitions and the agent they belong to at system architecture high level view.

When receiving the client request, the **Client** agent forwards the request to the **Mediator** agent with the **SendRequest** capability. Upon the reception of client request, the **Mediator** agent analyses whether it can realize this request with the **AnalyseClientRequest** capability. If it is the case, it will decompose this request into a set of sub-requests with the **DecomposeClientRequest** capability and then asks the **Matchmaker** agent for the relevant information of the corresponding capabilities with the **SendCapabilityInformationRequest** capability. The **Matchmaker** agent queries its knowledge (about the subscribed capability providers that can provide the relevant capability) with the **QueryAvailability** capability. If the **Matchmaker** agent cannot find any capability providers that can provide the relevant capability, the **Mediator** agent handles this negative answer with the **HandleMMRefusal** capability. Otherwise, it records the relevant information of corresponding capabilities with the **RecordCapabilityInformation** capability and then directly communicates with the corresponding capability provider. It sends sub-requests to corresponding capability providers with the

Table 2. Capabilities relating to sub-agents inside a capability provider

Capability Name	Informal Definition	Agent
QueryKB	Query the knowledge base to find the required capability	Connector
SendToBeTranslatedRequest	Send a request to an adapter	Connector
TranslateClientRequest	Translate client request to component language	Adapter
InvokeCOTSCapability	Invoke capability provided by COTS component	Adapter
InvokeCapability	Invoke custom capability	Connector
TranslateCOTSResult	Translate the result given by COTS component to client language	Adapter
SendResult	Send result to Mediator	Connector
HandleCapabilityFailure	Handle the failure occur during the capability invocation	Connector

SendSubRequest capability. It records the answers from the capability providers for each sub-request and integrates them to formulate the final result before sending it to the **Client** agent through **RecordSubResult**, **IntegrateSubResult** and **SendIntegratedResult** capabilities. The **Mediator** agent can also handle the negative answer from capability providers with its **HandleSubRequestFailure** capability. The **Client** agent records the result sent by the **Mediator** agent with **RecordResult** capability. If the **Mediator** agent cannot realize the client request, the **Client** agent handles this situation using its **HandleFailure** capability.

Capabilities relating to the sub-agents inside a capability provider are listed in Table 2. When receiving request from the **Mediator** agent, the **Connector** agent queries its knowledge about the capabilities offered with the **QueryKB** capability. If the capability requested is offered by the component, it communicates with the **Adapter** agent in order to invoke the corresponding capability by using **SendToBeTranslatedRequest**. Otherwise, it invokes directly the corresponding custom capabilities. Upon request reception, the **Adapter** agent is in charge of translating the request into the specific language of the component, consequently acting as a parser; **TranslateClientRequest**, **InvokeCOTSCapability**, and **TranslateCOTSResult** are in charge of the execution of the function. The **Connector** agent uses **HandleCapabilityFailure** to handle the situation when the capability provider cannot realize the requested capability due to (i) unknown capability, (ii) request translation failure, or (iii) capability invocation failure.

Table 3 lists the capabilities of the Subscription and Unsubscription patterns. In our case, it concerns the **Connector** and **Matchmaker** agents. The **Connector** agent sends the subscription request to the **Matchmaker** agent with the **SendSubscriptionRequest** capability. When receiving such a request, the **Matchmaker** agent decides and then sends its answers to capability provider through **SendSubscriptionRequestDecision** capability. There are two possible answers: 1) the subscription request cannot be subscribed; 2) the subscription request can be subscribed. The **Connector** agent processes these answers

Table 3. Capabilities of Subscription and Unsubscription patterns

Capability Name	Informal Definition	Agent
SendSubscriptionRequest	Send a request to subscribe capability	Connector
SendSubscriptionRequest-Decision	Decide and send a decision about the subscription received	Matchmaker
RecordCPSubscription	Record the capability provider subscription	Matchmaker
HandleMMSubscription-Refusal	Handle a negative answer from Matchmaker	Connector
RecordMMSubscription-Acceptance	Record a positive answer from Matchmaker	Connector
SendUnsubscriptionRequest	Send a request to unsubscribe capability	Connector
SendUnsubscriptionRequest-Decision	Decide and send a decision about the unsubscription received	Matchmaker
DeleteCPSubscription	Delete the existed capability provider subscription	Matchmaker
HandleMMUnsubscription-Refusal	Handle a negative answer from Matchmaker	Connector
RecordMMUnsubscription-Acceptance	Record a positive answer from Matchmaker	Connector

with `HandleMMSubscriptionRefusal` and `RecordMMSubscriptionAcceptance` capabilities, respectively. If the subscription request is accepted, the `Matchmaker` agent records this subscription using the `RecordSPSSubscription` capability. The other capabilities listed in Table 3 are used in the case of unsubscription.

3.3 Communicational Dimension

The communicational dimension models, in a temporal manner, events exchanged into the system (events allow agents to interact). Graphically, an AUML sequence diagram is used to represent this “event exchange” between agents.

Figure 2 shows the communication diagram of our multi-agent system architecture at the high level view. When the `Client` agent forwards a client request (`ClientRequestSent`) to the `Mediator` agent, the `Mediator` agent receives and analyses the request. In case that the `Mediator` agent cannot answer the request, it sends a failure message (`FailureAnalyzedClientRequestSent`) to the `Client` agent. Otherwise, it processes the request decomposing. If the `Mediator` agent fails to decompose the request, it then sends a failure message (`FailureDecomposedClientRequestSent`) to the `Client` agent. Otherwise it asks the `Matchmaker` agent for the relevant information of the corresponding capabilities (`CapabilityInformationRequestSent`). The `Matchmaker` agent may alternatively answer with a denial (`MMRefusalSent`) or information of the relevant capability (`CapabilityInformationSent`). Respectively, the `Mediator` agent will send a failure message (`FailureSent`) to the `Client` agent or send sub-request (`SubRequestSent`) to relevant capability provider. There are two possible answers from capability provider: a negative (`FailureSent`) or a positive

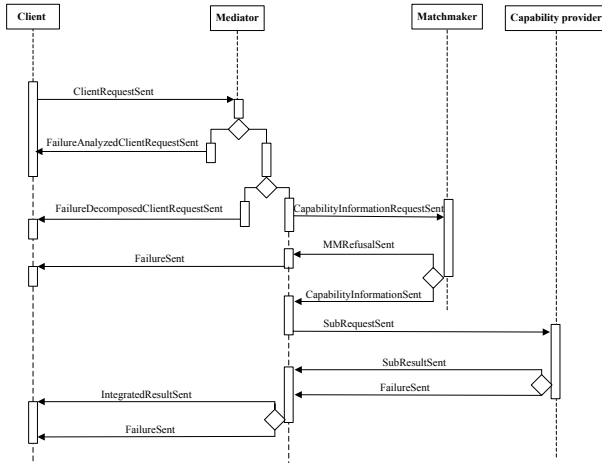


Fig. 2. Communication diagram of our MAS architecture at the high level view

(SubResultSent). If the Mediator agent gets a negative answer from capability provider, it will send a failure message (FailureSent) to the Client agent. Otherwise, the integrated result (IntegratedResultSent) will be sent to the Client agent.

The communication diagram relating to sub-agents inside a capability provider is depicted in Figure 3. When the Connector agent receives request from outside, if the requested capability is offered by a component, it sends the request to the Adapter agent to translate it into a language that is understood by components (RequestToBeTranslatedSent). If the Adapter agent cannot translate the request, it sends a failure message to the Connector agent (FailureTranslatedRequestSent). Otherwise, the translated request will be sent to COTS product (RequestTranslatedSent). If the component capability is successfully invoked, the answer from the component is sent back to the Adapter agent to translate it into the Client language (CapabilityResultSent). Otherwise, a failure message will be sent to the Adapter agent that will forward it to the Connector agent (FailureSent). In the case that the capability requested is not offered by the component, the Connector agent sends the request to invoke the custom capability (RequestSent) and two possible answers from the invoked capability: a failure message (FailureSent) or a result (CapabilityResultSent).

The communication diagram for the Subscription and Unsubscription patterns is depicted in Figure 4. In the case of subscription, the Connector agent send a subscription request containing the information about the capability it provides (SubscriptionSent) to the Matchmaker agent. When receiving this request, the Matchmaker agent may answer with an (AcceptedSubscriptionSent) acceptance message or a denial one (RefusedSubscriptionSent). A similar process is applied in the case of unsubscription.

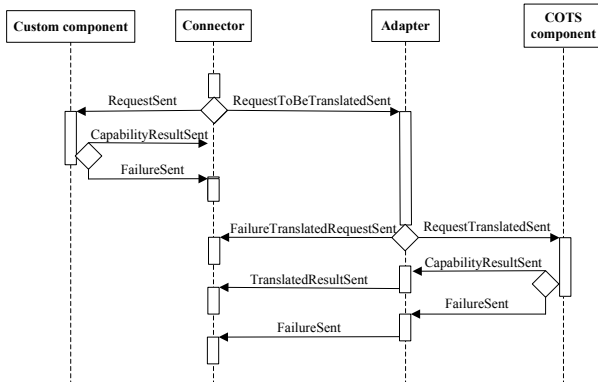


Fig. 3. Communication diagram relating to sub-agents inside a capability provider

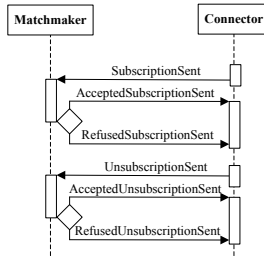


Fig. 4. Communication diagram of Subscription and Unsubscription patterns

3.4 Dynamic Dimension

The dynamic dimension models the synchronization mechanisms between events and capabilities. A capability can be invoked by an event and can create new events so that, graphically, an extended activity diagram called *dynamic diagram* in [9] is used to model their synchronization and relationships.

In a dynamic diagram, each agent constitutes a swimlane of the diagram. The capability is represented in a round-corner box and placed in the swimlane corresponding to the agent that it belongs to. An internal event is represented by a dashed arrow and an external event by a solid arrow. An event may be handled by alternative capabilities that are enclosed in a box. Synchronization and branching are represented as usual.

The dynamic diagram relating to different agents at the high level view of our MAS architecture is depicted in Figure 5 and the dynamic diagram relating to the sub-agents inside a capability provider is illustrated in Figure 6 while Figure 7 represents the dynamic diagram of the Subscription and Unsubscription pattern. These dynamic diagrams show the synchronization between the agents' capabilities identified in the intentional dimension (see Section 3.2) and the events presented in the communicational dimension (see Section 3.3).

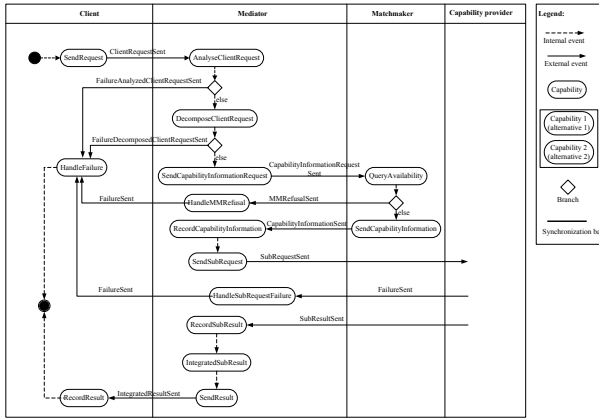


Fig. 5. Dynamic diagram of our MAS architecture at the high level view

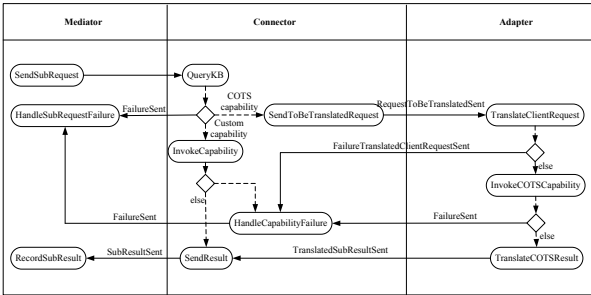


Fig. 6. Dynamic diagram relating to sub-agents inside a capability provider

4 Validation

The validation of the proposed architecture is extremely difficult to quantify. A same case study should be performed using the methodology proposed in [17], once with the architectural pattern, once without. The result could then be evaluated on the basis of development time, cost, the application performance on general and particular aspects, etc. Such an experiment would nevertheless be polluted by external factors. More specifically, we could include an evaluation of a small sized project on the basis of software metrics. Indeed, to focus on the contribution of such a design-pattern we point to “structural complexity evaluation”. Indeed, structural complexity focuses on MAS architecture and agent relationships, features that should be enriched using a coordination pattern. Due to the poorness of literature concerning agent-oriented software metrics evaluating structural complexity, we point to the use of existing object-oriented ones. As a preliminary tests suite, we claim for the use of the metrics proposed by [4]. Due to a lack of space and time, this evaluation is left for future work.

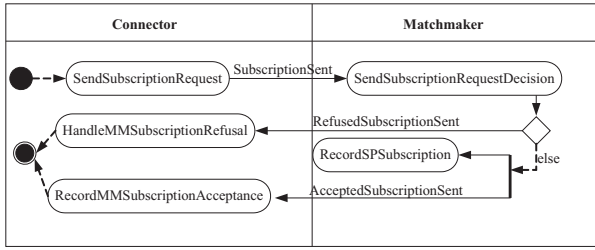


Fig. 7. Dynamic diagram of Subscription pattern (Unsubscription is similar)

5 Conclusion

CBSD presents new challenges tending to limit their use. The architecture presented here addresses some of these issues using a wrapper acting as an abstraction layer between the client's (high level) requests, their functional decomposition and the coordination of execution by the components. This allows software engineers to focus on requirements models on the one side and on (standardized) interfacing with the MAS on the other in order to adequately tune and configure the architecture. This brings further formalization into CBSB.

References

1. Altendorf, E., Hohman, M., Zabicki, R.: Using j2ee on a large, web-based project. *IEEE Software* 19(2), 81–89 (2002)
2. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8(3), 203–236 (2004)
3. Brown, A.W., Wallnau, K.C.: The current state of component-based software engineering. *IEEE Software*, 37–46 (1998)
4. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. *IEEE Trans. Software Eng.* 20(6), 476–493 (1994)
5. Davis, A., Zhang, D.: A comparative study of dcom and soap. In: *ISMSE*, pp. 48–55. IEEE Computer Society, Los Alamitos (2002)
6. Fang, G.-M., Lin, J.-M.: An architecture for multi-agent cots software integration systems. In: *ICPADS* (2), pp. 580–584. IEEE Computer Society, Los Alamitos (2005)
7. Felber, P., Narasimhan, P.: Experiences, strategies, and challenges in building fault-tolerant corba systems. *IEEE Trans. Computers* 53(5), 497–511 (2004)
8. FIPA. Agent uml (2010), <http://www.auml.org/>
9. Kolp, M., Faulkner, S., Wautelet, Y.: Social structure based design patterns for agent-oriented software engineering. *IJIIT* 4(2), 1–23 (2008)
10. Kontio, J.: A cots selection method and experiences of its use. In: *Proceedings of the 20th Annual Software Engineering Workshop, Maryland* (1995)
11. Kunda, D., Brooks, L.E.: Applying social-technical approach for cots selection. In: *Proceeding of the 4th UKAIS Conference, University of York* (1999)
12. Ncube, C., Maiden, N.A.M.: Pore: Procurement-oriented requirements engineering method for the component-based systems engineering development paradigm. In: *International Workshop on Component-based Software Eng.* (1999)

13. Pour, G.: Component-based software development approach: New opportunities and challenge. In: Proceedings Technology of Object-Oriented Languages, TOOLS 26, pp. 375–383 (1998)
14. Rose, L.C.: Risk management of COTS based systems development. In: Cechich, A., Piattini, M., Vallecillo, A. (eds.) Component-Based Software Quality. LNCS, vol. 2693, pp. 352–373. Springer, Heidelberg (2003)
15. Poggi, A., Rimassa, G., Turci, P.: What agent middleware can (and should) do for you. *Applied Artificial Intelligence* 16(9) (2002)
16. Vigder, M.R., Dean, J.C.: An architectural approach to building systems from cots software components. In: in Proc. of the 22nd Annual Soft. Eng. Workshop (1997)
17. Wautelet, Y., Kiv, S., Kolp, M.: An iterative process for component-based software development centered on agents. *T. Computational Collective Intelligence* (2011)
18. Yu, E.: *Social Modeling for Requirements Engineering*. MIT Press, Cambridge (2011)

A Multi-Agent System for Game Trading on the B2B Electronic Market

Ana Petric and Gordan Jezic

University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Zagreb, Croatia
{ana.petric,gordan.jezic}@fer.hr

Abstract. In order to provision content-based services to their users, service providers need to purchase content distribution rights from various content providers. Sometimes the creation phase of certain content categories can last longer than planned so content providers deliver the content to service providers with delays. The delayed delivery can cause financial damages and inconveniences to service providers which have already announced the appearance of new content to their users. In this paper we propose a multi-agent system which is used to model an electronic market for trading the rights to distribute video and computer games. On the proposed market, a service provider agent uses the multi-attribute reputation decision making mechanism to purchase content distribution rights from content provider agents. Several sets of experiments were conducted to demonstrate the efficiency of the multi-attribute reputation decision making mechanism.

Keywords: content trading, B2B content e-market, reputation tracking auctions, multi-agent system.

1 Introduction

The ability to transfer various video and audio data as well as other forms that carry certain information into digital form so it can be deployed across multiple technologies is considered to be one of the fundamental enablers of convergence between different domains (i.e., telecommunications, the Internet, information technology, broadcasting and media) that are involved in the content service provisioning process [3,13]. The appearance of digital content was followed by the development of (digital) content market while the advances in the ICT (Information and communication technologies) industry enabled the development of (digital) content electronic market (e-market).

Service providers (i.e., network operators, internet service providers, cable television operators) recognized the potential of the content e-market and started to develop new business models for provisioning content-based services to their users [15,6]. The term content encompasses movies, songs, games, news, images and text, in other words data and information within various fields [12]. Since

content production is not in service providers' primary business domain, in order to provision content-based services, they have to buy content distribution rights from various content providers (i.e., media companies, copyright holders) on the content e-market [14]. Content providers sell content distribution rights to service providers on the Business-to-Business (B2B) content e-market, while service providers provision content-based services to users on the Business-to-Consumer (B2C) content e-market [7,10].

Games are one of the youngest content categories while the (digital) games market is considered to be one of the most advanced content markets when it comes to digital distribution and exploitation [11]. Some content categories, such as games, have longer and/or expensive creation phases so content providers can sell content distribution rights on the B2B content e-market during the creation phase even before the content is finished (e.g., game distribution rights can be sold once the story line has been created and the visuals of the game have been designed). If the creation phase lasts longer than anticipated by the creation deadlines, games can be delivered to service providers with delays causing financial damages and inconveniences to service providers.

The paper is organized as follows. Section 2 presents the architecture of a multi-agent system for game trading while section 3 presents the settings of the conducted experiments and the obtained results. Section 4 concludes the paper and gives an outline for future work.

2 Architecture of the Multi-Agent System

Figure 1 shows the architecture of the proposed multi-agent system for game trading that is defined as $MAS_{gt} = \{SPA, CPA_1, \dots, CPA_i, \dots, CPA_M\}$. The Service Provider Agent SPA represents a network operator which acts as a service provider on the content e-market while M Content Provider Agents $CPAs$ represent game publishing companies which publish (and produce) games and act as content providers on the B2B content e-market. On the B2C content e-market, the SPA provides game-based services to users which are represented by their User Agents (UAs).

2.1 Content Provider Agent

The content provider agent CPA_i sells game distribution rights to the SPA . The CPA_i is defined as follows $CPA_i = \{X_{cpa_i}, dem, r_c(i)\}$ where X_{cpa_i} represents the current CPA_i 's games offer, dem stands for the delivery evaluation mechanism which is used to calculate the on-time delivery probability for a certain game, while the $r_c(i)$ is the CPA_i 's risk taking policy which represents the lowest acceptable value of the on time delivery probability defined by the game publishing company's business strategies.

The CPA_i receives a Call For Proposals (CFP) from the SPA and decides will it participate in an auction held by the SPA . If the CPA_i does not have at its disposal any games ready for distribution, it calculates the on-time delivery

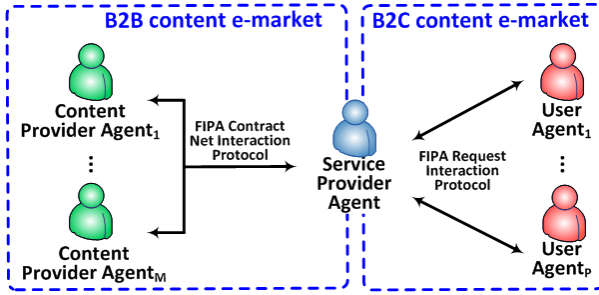


Fig. 1. Architecture of the multi-agent system for games trading

probability $p(t_c(\mathbf{x}_i))$ that the creation phase of a certain game $\mathbf{x}_i \in \mathbf{X}_{cpa_i}$ will be completed by the deadline t_c which is defined in the CFP. If the probability $p(t_c(\mathbf{x}_i))$ is higher than its risk taking policy $r_c(i)$, the CPA_i sends an offer \mathbf{x}_i .

2.2 Service Provider Agent

The SPA is defined as follows $SPA = \{th, gc, cem, rtm, mrdm\}$, where the th is the trading history with $CPAs$, the gc contains information about prior users' games consumption, the cem is the content evaluation mechanism, the rtm is the reputation tracking mechanism while the $mrdm$ is the multi-attribute reputation decision making mechanism. The information from the th is gathered from prior transactions with $CPAs$ and used by the rtm . The information contained in the gc is gathered from users' consumption behaviour on the B2C e-content market and used to determine reserve and aspiration values used by the cem .

After analysing the information from the gc , the SPA determines relevant game attributes $\mathbf{A} = \{a_1, \dots, a_j, \dots, a_n\}$ as well as their reserve and aspiration values necessary for the evaluation of games offered by $CPAs$. Reserve value x_j^r marks the lowest value of an attribute a_j that is acceptable to the SPA while the aspiration value x_j^a is the highest value of an attribute a_j that the SPA is interested in. The SPA starts an auction by sending CFPs to $CPAs$. The SPA acts as a buyer while $CPAs$ act as sellers in the auction. The required delivery deadline t_c as well as a list of relevant attributes \mathbf{A} and their reserve values $\mathbf{x}^r = \{x_1^r, \dots, x_j^r, \dots, x_n^r\}$ are sent within the CFP. The SPA receives $m \leq M$ offers $\mathbf{X}_m = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_m\}$ from $CPAs$ that decided to participate in the auction where the CPA_i offers to sell distribution rights of the game $\mathbf{x}_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{in}\}$.

Content Evaluation Mechanism. The cem [9] is represented with a tuple $\langle \mathbf{X}_m, w_c, u_c, d, u \rangle$, where each offer $\mathbf{x}_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{in}\}$ from the set of offers \mathbf{X}_m is described with n attributes, $w_c = \{w_{c1}, \dots, w_{cj}, \dots, w_{cn}\}$ is a set of weights that determines the importance of each attribute from \mathbf{x}_i , u_c is a utility function that calculates the SPA 's utility of CPA_i 's offer \mathbf{x}_i ; d is a

deviation function that calculates the *SPA*'s additional benefit of *CPA*_{*i*}'s offer \mathbf{x}_i ; u is a utility function that calculates the total utility of the offer \mathbf{x}_i and it is defined as follows:

$$u(\mathbf{x}_i) = (1 - w_{c,bonus})u_c(\mathbf{x}_i) + w_{c,bonus}d(\mathbf{x}_i). \tag{1}$$

The primary objective of the $u(\mathbf{x}_i)$ is to maximize content utility $u_c(\mathbf{x}_i)$ of the offered content \mathbf{x}_i while the secondary objective is to maximize additional benefits $d(\mathbf{x}_i)$ that an offer brings. The weight of additional benefits $w_{c,bonus}$ should be low enough to prevent a *CPA* with an average offer from manipulating the auction outcome in its favour, but at the same time it should reward a *CPA* with a very good offer which also brings additional benefits to the *SPA*.

The content utility function $u_c(\mathbf{x}_i)$ is an additive scoring function that assumes the existence of mutual *preferential independence* between attributes [2]. It is defined as follows:

$$u_c(\mathbf{x}_i) = \sum_{j=1}^n w_{cj}u_c(x_{ij}), \text{ where } \sum_{j=1}^n w_{cj} = 1 \tag{2}$$

$$u_c(x_{ij}) = \begin{cases} \frac{x_{ij}-x_j^r}{x_j^a-x_j^r}, & x_j^r \neq x_j^a \ \& \ x_j^r \leq x_{ij} \leq x_j^a \\ \text{N.A.}, & x_{ij} < x_j^r \\ 1, & x_{ij} > x_j^a \end{cases} \tag{3}$$

Value N.A. in (3) marks a non-acceptable value for an attribute. Values offered higher than the aspiration value are acceptable, but their utility cannot be higher than 1. The positive deviation function $d(\mathbf{x}_i)$ compares an offer \mathbf{x}_i placed by the *CPA*_{*i*} with the aspiration offer $\mathbf{x}^a = \{x_1^a, \dots, x_j^a, \dots, x_n^a\}$ taking into consideration highest possible attribute values x_j^{max} as follows:

$$d(\mathbf{x}_i) = \sqrt{\sum_{j=1}^n d(x_{ij})^2}, \quad d(x_{ij}) = \begin{cases} w_{cj} \frac{x_{ij}-x_j^a}{x_j^{max}-x_j^a}, & \text{if } x_{ij} > x_j^a \ \& \ x_j^a \neq x_j^{max} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Reputation Tracking Mechanism. The *rtm* [8] is represented with a tuple $\langle y, w_r, V, r \rangle$, where $y = \{y_1, y_2, y_3\}$ is a set of attributes used to calculate *CPAs*' reputation, $w_r = \{w_{r1}, w_{r2}, w_{r3}\}$ is a set of weights that determines the importance of each attribute from y for the *SPA*, V is the *SPA*'s sliding window which is used to store information regarding the last v auctions the *SPA* held; and r is a utility function that calculates *CPAs*' reputation.

Three attributes relevant for calculating *CPA*_{*i*}'s reputation are: y_1 – the average utility of games the *CPA*_{*i*} offers; y_2 – the share of total transactions carried out by the *CPA*_{*i*}; y_3 – the share of financial damages inflicted on the *SPA* due to *CPA*_{*i*}'s late delivery. The formulas for calculating reputation are as follows:

$$r(CPA_i) = \sum_{l=1}^3 w_{rl}y_l = w_{r1} \frac{1}{a_i} \sum_{k=1}^v \frac{a_i^k u(\mathbf{x}_i^k)}{u(\mathbf{x}_{w_i n}^k)} + w_{r2} \frac{\sum_{k=1}^v c_i^k n_k p_k x_{i1}^k}{\sum_{k=1}^v n_k p_k x_{w_i n}^k} + w_{r3} \left(1 - \frac{\sum_{k=1}^v c_i^k d_i^k}{\sum_{s=1}^m \sum_{k=0}^v c_s^k d_i^k} \right), \text{ where } : \sum_{l=1}^3 w_{rl} = 1 \tag{5}$$

$$a_i = \sum_{k=1}^v a_i^k, \quad a_i^k = \begin{cases} 1, & \text{if } CPA_i \text{ participated in auction } k \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$c_i^k = \begin{cases} 1, & \text{if } CPA_i \text{ won in auction } k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$d_i^k = n_{k1} p_k x_{i1}^k \quad (8)$$

In (5), x_{win}^k denotes the winning offer in auction k , n_k denotes how many times has the *SPA* sold a certain game, p_k denotes the selling price of a game, while x_{i1}^k is the percentage of the income from selling games that the *SPA* will get, a_i^k in (6) denotes CPA_i 's participation in auction k ; c_i^k in (7) denotes whether the CPA_i won auction k . In (8), d_i^k denotes the estimated financial damage inflicted to the *SPA* due to the CPA_i 's late delivery while n_{k1} denotes the estimated number of users that would have bought the game in the period between the arranged and actual delivery date.

Multi-attribute Reputation Decision Making Mechanism. The *mrdm* is defined as a tuple $\langle cem, rtm, w_t, t \rangle$ where *cem* and *rtm* are previously described content evaluation mechanism and reputation tracking mechanism, respectively, $w_t = \{w_{t1}, w_{t2}\}$ denotes a pair of weights where w_{t1} and w_{t2} determine the importance of *cem* and *rtm*, respectively, and t is the total score function. The winner of the *SPA*'s auction is the CPA_i with the highest total score $t(\mathbf{x}_i)$ which is determined as follows:

$$t(\mathbf{x}_i) = w_{t1}u(\mathbf{x}_i) + w_{t2}r(cpa_i), \quad \text{where } w_{t1} + w_{t2} = 1 \quad (9)$$

3 Case Study: Games Trading

The proposed multi-agent system was implemented using the Java Agent Development Framework (JADE¹) and it consists of one *SPA* and 8 *CPAs*. The *SPA* conducts auctions by using the *FIPA Contract Net Interaction Protocol*².

In this case study publishing companies announce the appearance of new games roughly one month earlier while the service provider offers few new games to its users every day what comes up to 25 games in one week. So, each week the *SPA* conducts 25 sequential auctions and purchases the rights to distribute games which the service provider is going to start offering to his users during one week in the next month. Since the service provider announces the appearance of new games to its users it is important that the games are delivered on time.

Before the beginning of an auction, the *SPA* determines relevant attributes and their appropriate (i.e., reserve and aspiration) values which are going to be used for the evaluation of offered games. Those attributes are: a_1 - a percentage

¹ <http://jade.tilab.com>

² <http://www.fipa.org/specs/fipa00029/index.html>

Table 1. Range of attribute values and SPA’s attribute valuations

	a_1	a_2	a_3	a_4	a_5
minimum value (x_j^{min})	0	t_1	0	0	1
maximum value (x_j^{max})	100	t_8	100	100	100
weight (w_j)	0.30	0.25	0.20	0.15	0.10
reservation value (x_j^r)	15	t_1	50	15	30
aspiration value (x_j^a)	40	t_8	90	45	75

Table 2. Content providers’ risk taking policies

<i>agent</i>	CPA_1	CPA_2	CPA_3	CPA_4	CPA_5	CPA_6	CPA_7	CPA_8
$r_c(i)$	0.6	0.65	0.7	0.8	0.55	0.6	0.65	0.7

of income from each sold game that the SPA will get, a_2 - game type, a_3 - a percentage of mobile phone types which have the necessary software and hardware support to run the offered game, a_4 - a percentage of income from selling various advanced game options that the SPA will get, and a_5 - the time period during which the SPA has the right to sell the game.

Table 1 contains the minimum (i.e., worst) and maximum (i.e., best) possible attribute values as well as SPA’s valuations (i.e., weights), reservation and aspiration values for each attribute. Minimum and maximum values as well as SPA’s valuations of all attributes remained the same during the experiment. We assume that the game will not be interesting to users longer than three months so we set the x_5^{max} on 100 days even though x_5^{max} can actually be indefinite. Game types are taken from Ovi³: t_1 - action, t_2 - arcade, t_3 - education, t_4 - card & casino, t_5 - adventure, t_6 - puzzle and trivia, t_7 - sports, and t_8 - strategy.

After the CPA_i receives a CFP it determines the probability that the game x_i will be completed on time. If the probability is higher than its risk taking policy $r_c(i)$, the CPA_i offers the game x_i to the SPA. In this case study the CPA_i ’s on time delivery probability for a game x_i in the auction k is determined by (10) while the CPAs’ risk taking policies are listed in Table 2.

$$p_k(t_c(x_i)) = p_{k-1}(t_c(x_i)) + random[-0.05, 0.05] \tag{10}$$

Aspiration and reserve values are determined from game sales records. Since we do not model users behaviour, in this experiment reserve and aspiration values were chosen randomly from intervals $[0.95x_j^r(nominal), 1.05x_j^r(nominal)]$ and $[0.95x_j^a(nominal), 1.05x_j^a(nominal)]$, respectively, after each set of 25 auctions. The values $x_j^r(nominal)$ and $x_j^a(nominal)$ are given in Table 1. The SPA accepted offers containing all game types. The utility of a certain game type t_j was equal to its market share in the last period δt (i.e. last week) as follows:

$$u_c(x_{i2}) = n_{t_j}(\delta t) / \sum_{j=1}^8 n_{t_j}(\delta t), \tag{11}$$

³ <https://store.ovl.com/#/games>

while the number of games sold in one week $n_{t_j}(\delta t)$ for each game type t_j was chosen randomly from the interval $[0.8n_{t_j}(nominal), 1.2n_{t_j}(nominal)]$ where $n_{t_j}(nominal)$ for all games type was 12500.

CPAs also calculate their reserve and aspiration values from game sales records so those values do not differ significantly from the values calculated by the SPA. CPAs' reserve and aspiration values are chosen randomly from intervals $[0.95x_j^r(nominal), 1.05x_j^r(nominal)]$ and $[0.95x_j^a(nominal), 1.05x_j^a(nominal)]$, respectively, while their nominal values are given in Table 3. The CPAs offer values that are chosen randomly between CPAs' reserve and aspiration values.

Table 3. Content provider agents' weights, reserve and aspiration values

Attribute	a_1	a_2	a_3	a_4	a_5	a_1	a_2	a_3	a_4	a_5
	Content Provider Agent CPA_1					Content Provider Agent CPA_2				
w_{c_j}	0.35	0.21	0.15	0.20	0.09	0.34	0.19	0.145	0.21	0.115
x_j^r	37.5	t_1	38	90	45	36	t_1	34	88	41
x_j^a	12.5	t_8	24	60	85	14	t_8	16	61	86
	Content Provider Agent CPA_3					Content Provider Agent CPA_4				
w_{c_j}	0.33	0.18	0.175	0.19	0.125	0.36	0.24	0.16	0.175	0.085
x_j^r	38	t_1	33.5	92	49	37.5	t_1	34.5	87	43
x_j^a	13	t_8	14	58	90	15.5	t_8	13	63	91
	Content Provider Agent CPA_5					Content Provider Agent CPA_6				
w_{c_j}	0.345	0.22	0.14	0.165	0.13	0.335	0.21	0.155	0.185	0.125
x_j^r	39	t_1	33	85	47	36.5	t_1	37	83	46
x_j^a	16	t_8	12	54	88	13	t_8	11	57	89
	Content Provider Agent CPA_7					Content Provider Agent CPA_8				
w_{c_j}	0.36	0.23	0.13	0.15	0.13	0.375	0.19	0.21	0.115	0.11
x_j^r	38.5	t_1	35.5	94	44	37.5	t_1	38.5	92	48
x_j^a	13.5	t_8	15	67	87	14.5	t_8	16	59	90

Once the SPA chooses the winning CPA_i of auction k , whether the CPA_i will deliver the game on time or with a delay is determined as defined in Listing 1. In case the delivery delay occurs, the number of users $n_{k1}(\mathbf{x}_i)$ that would have bought the game \mathbf{x}_i in the period between the agreed and actual delivery time is calculated as follows:

$$n_{k1}(\mathbf{x}_i) = n_k(\mathbf{x}_i) * (1 - e^{(1-temp)}), \tag{12}$$

where the variable $temp$ is taken from Listing 1 while the total number of users $n_k(\mathbf{x}_i)$ that would have bought the game \mathbf{x}_i if it was delivered on time is determined by (13) where t_j is the game type of game \mathbf{x}_i .

$$n_k(\mathbf{x}_i) = \frac{\sum_{l=((k \bmod 25)-1) \cdot 25+1}^{(k \bmod 25) \cdot 25} n_l(\mathbf{x}_i^l) \cdot u(\mathbf{x}_i^l)}{|\ n_l(\mathbf{x}_i^l \mid x_{i2} = t_j) \ | \cdot \frac{1}{25} \sum_{l=((k \bmod 25)-1) \cdot 25+1}^{(k \bmod 25) \cdot 25} u_l(\mathbf{x}_i^l | i=winner)} } \tag{13}$$

Listing 1. The on time delivery occurrence

```

temp = random[0, 1]
if temp < pk(xi) then
    on time delivery
else
    delivery delay
end if

```

The total SPA’s income from selling a game \mathbf{x}_i bought in auction k is:

$$income_{r/nr,k}(\mathbf{x}_i) = (n_k - n_{k1}) \cdot p_k \cdot x_{i1} \tag{14}$$

where the label r/nr specifies whether the SPA took the CPA_{*i*}’s reputation into account (label r) when it determined the winner of the auction, or did not (label nr), p_k is the game unit price while the x_{i1} is the percentage of the total income that the SPA keeps from the game sales.

For each auction in the experiment we determined the winner of the auction by using the multi-attribute reputation decision making mechanism and calculated the income $income_r$ from the game sales by using (14). For each auction we also determined the winner by using just the content evaluation mechanism (i.e., the weight w_{t2} in (9) was set to 0 so the CPAs’ reputation was not taken into account) and calculated the accompanying income $income_{nr}$. The efficiency η of the reputation tracking mechanism is determined by comparing the previously defined incomes as follows:

$$\eta = \frac{\sum_{k=1}^T income_{r,k}(\mathbf{x}_i^k) - \sum_{k=1}^T income_{nr,k}(\mathbf{x}_j^k)}{\sum_{k=1}^T income_{nr,k}(\mathbf{x}_j^k)}, \forall k \text{ when } cpa_i \neq cpa_j. \tag{15}$$

where T is the total number of reputation tracking auctions in one experiment.

We conducted nine sets of experiments as follows. We changed the values of reputation attribute weights (i.e., w_{r1} , w_{r2} and w_{r3}) in each set and conducted 100 series of experiments per set. Each series of experiments is conducted with different total attribute weights (i.e., w_{t1} , w_{t2}). In the first series the weight w_{t2} is set to 0.01 and it is increased in each following series by 0.01. One series of experiments consists of 100 experiments while one experiment includes 2800 seal-bid multi-attribute reverse auctions held sequentially one after the other. Since the reputation in an ongoing auction was calculated from the trading history gathered from previous 300 auctions, the first 300 out of 2800 auctions did not take reputation into account when determining the winner of an auction but they were used to calculate CPAs’ reputations later. The remaining 2500 auctions took reputation into account when determining the winner of the auction. In the conducted experiments we tried to increase SPA’s income by reducing the damages inflicted to the SPA as a consequence of CPAs’ late deliveries. The emphasis was put on the reputation attribute y_3 by setting a higher value of reputation attribute weight w_{r3} .

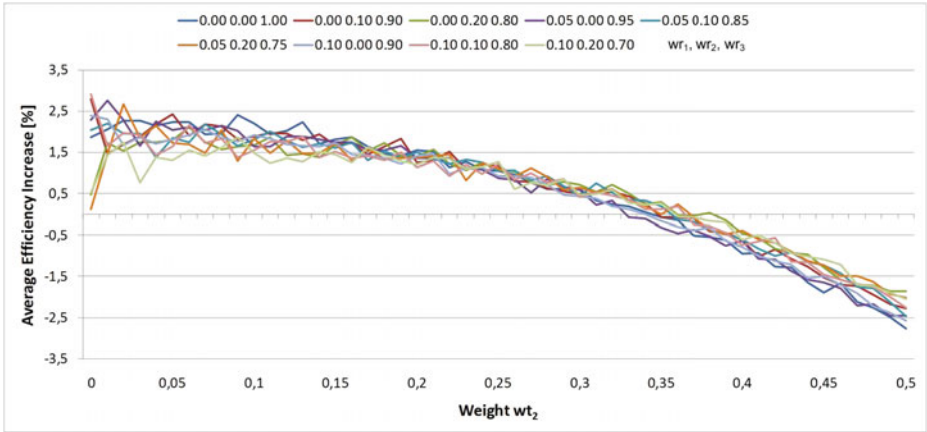


Fig. 2. Efficiency of the reputation tracking mechanism

In Figure 2 we can see the results of the conducted experiments. At the end of each experiment the efficiency of the reputation tracking mechanism is calculated by using (15). Since one series consists of 100 experiments, Figure 2 shows a mean value of efficiency for each series. The efficiency of the reputation tracking mechanism varies between 1.5 and 2.5% for weights $w_{t2} \in [0.05, 0.15]$ and continues to decrease afterwards. Approximately after the w_{t2} rises above 0.35 the reputation tracking mechanism becomes inefficient (i.e., $\eta < 0$).

4 Conclusion and Future Work

The presented multi-attribute reputation decision making mechanism enables service providers to negotiate the purchase of content distribution rights by taking into account content attributes as well as the content providers’ reputation. The reputation is based on: i) the utility of content providers’ offered content in previous auctions what gives a certain advantage to content providers which offer high quality content; ii) the ratio of incomes realized from selling content of a certain content provider and service provider’s total income; and iii) the ratio of damages that a certain content provider caused with its delayed content deliveries and the total damages caused by all content providers. We implemented a multi-agent system in which the Service Provider Agent uses the proposed multi-attribute reputation decision making mechanism to purchase game distribution rights from Content Provider Agents on the B2B e-content market. From the conducted experiments we can see that the efficiency of the proposed mechanism varies between 1.5 and 2.5% when the importance of the reputation in the decision making process varies between 5 and 15% while the mechanism becomes inefficient when the importance of the reputation rises above 35%.

For future work we plan to model several strategies which content providers will use to prepare offers and continue our work with modelling users behaviour to get a complete content electronic market.

Acknowledgments. The authors acknowledge the support of research project "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Bang, Y.-H.: Nom 2.0: innovative network operations and management for business agility. *IEEE Communications Magazine* 46(3), 10–16 (2008)
2. Bichler, M.: An experimental analysis of multi-attribute auctions. *Decision Support Systems* 29(3), 249–268 (2000)
3. Hanrahan, H.: *Network Convergence: Services, Applications, Transport, and Operations Support*. John Wiley & Sons, Inc., New York (2007)
4. LeClerc, M.: Swimming with the sharks. *Ericsson Business Review* 2(3), 18–22 (2007)
5. Olla, P., Patel, N.V.: A value chain model for mobile data service providers. *Telecommunications Policy* 26(9-10), 551–571 (2002)
6. Petersson, J., Strömberg, B.: The key to value-adding the connected home. *Ericsson Business Review* 4(1), 40–42 (2009)
7. Petric, A.: A Multi-Agent System for Content Trading in Electronic Telecom Markets Using Multi-Attribute Auctions. In: Simperl, E. (ed.) 10th International Conference on Electronic Commerce ICEC 2008 Proceedings of the Doctoral Consortium, pp. 1–5 (2008)
8. Petric, A., Jezic, G.: Reputation tracking procurement auctions. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 825–837. Springer, Heidelberg (2009)
9. Petric, A., Jezic, G.: Multi-attribute auction model for agent-based content trading in telecom markets. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6276, pp. 261–270. Springer, Heidelberg (2010)
10. Podobnik, V., Petric, A., Trzec, K., Jezic, G.: Software Agents in New Generation Networks: Towards the Automation of Telecom Processes. In: *Knowledge Processing and Decision Making in Agent-Based Systems*, pp. 71–99 (2009)
11. Digest, S., Rightscom, Goldmedia, Sigle, C.H.: *Interactive content and convergence: Implications for the information society*. Screen Digest Limited, London (2006)
12. Subramanya, S., Yi, B.K.: Utility Model for On-Demand Digital Content. *Computer* 38, 95–98 (2005)
13. Wu, I.: Canada, south korea, netherlands and sweden: regulatory implications of the convergence of telecommunications, broadcasting and internet services. *Telecommunications Policy* 28(1), 79–96 (2004)

Experimental Comparison of Resampling Methods in a Multi-Agent System to Assist with Property Valuation

Tadeusz Lasota¹, Tomasz Łuczak², and Bogdan Trawiński²

¹ Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
Ul. Norwida 25/27, 50-375 Wrocław, Poland

² Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{tomasz.luczak,bogdan.trawinski}@pwr.wroc.pl,
tadeusz.lasota@up.wroc.pl

Abstract. A new version of the a multi-agent system to aid in real estate appraisal, called MAREA-2, was introduced. The system is being developed using Java Spring Framework and is intended for industrial application in cadastral information centres. The major part of the study was devoted to investigate the performance of *Bagging*, *Subbagging*, and *Repeated cross-validation* models. The overall result of our investigation was that the majority of models created using resampling techniques provided better or equivalent accuracy than the experts' method employed in reality. It confirms that automated valuation models can be successfully incorporated into the multi-agent system and be utilized to support appraisers' work.

Keywords: bagging, subbagging, multi-agent systems, property valuation.

1 Introduction

We have been investigating techniques for developing a multi-agent system to aid in real estate appraisal devoted to information centres maintaining cadastral systems with digital cadastral maps and property sales transaction registries. Due to the substantial dispersion in Poland, these systems are located in district local self-governments as well as in the municipalities of bigger towns, and there are above 400 such centres all over the country. All three systems together could create a complex data source for real estate appraisers accessible through Internet (Fig. 1). At present the information centres are the place which appraisers contact when they start evaluating properties. Moreover, the appraisers are obliged to deliver the results of their estimates to the governmental registry of real estate transactions, therefore the actual values and prices of properties are available in the registry.

Bagging, which stands for **bootstrap aggregating**, devised by Breiman [4] belongs to the most intuitive and simplest ensemble algorithms providing a good performance. Diversity of learners is obtained by using bootstrapped replicas of the training data. That is, different training data subsets are randomly drawn with replacement from the original base dataset. So obtained training data subsets, called also bags, are used then

to train different classification or regression models. Finally, individual learners are combined through algebraic expressions. The classic form of bagging is the n -out-of- n with replacement bootstrap where the number of samples in each bag equals to the cardinality of a base dataset and as a test set the whole original dataset is used. In order to achieve better computational effectiveness less overloading techniques were introduced which consisted in drawing from an original dataset smaller numbers of samples, with or without replacement. The m -out-of- n without replacement bagging, where at each step m observations less than n are distinctly chosen at random within the base dataset, belongs to such variants. This alternative aggregation scheme was called by Bühlmann and Yu [5] subagging for **subsample aggregating**. In the literature the resampling methods of the same nature as subagging are also named Monte Carlo cross-validation [13] or repeated holdout [3]. In turn, subagging with replacement was called by Biau et al. [2] moon-bagging, standing for **m-out-of-n bootstrap aggregating**.

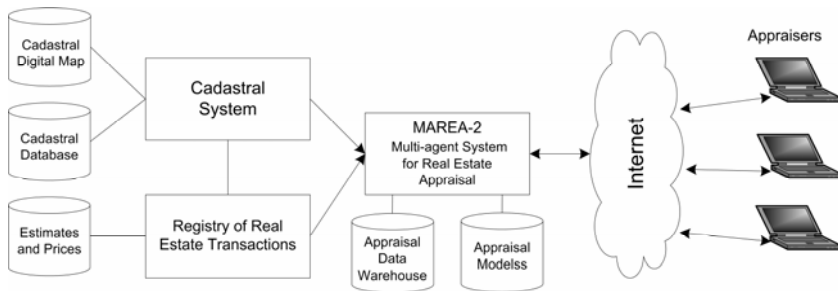


Fig. 1. Information systems to assist with real estate appraisals

In our previous work [11] we introduced the concept of a multi-agent system for real estate appraisals, called MAREA, employing six different machine learning algorithms. The design of the MAREA system was then extended to include aggregating agents, which applying the bagging procedure created homogeneous ensemble models [12]. In the next paper we presented our study on the application of heuristic optimization algorithms to construct heterogeneous bagging ensembles based on the results of fuzzy algorithms applied to real-world cadastral data [9]. The description of the MAREA system can be found in [1].

The first goal of the current contribution is to present a new version of the system, named MAREA-2, which is being developed using Java Spring Framework and is intended for industrial application in cadastral information centres in Poland.

The second goal of the paper is to examine the usefulness of machine learning algorithms taken from WEKA software in the form of Java classes to create ensemble models providing better performance than their single base models as well as the routine pairwise comparison method applied by professional appraisers. Due to the fact that data processing and learning agents may operate in an asynchronous off-line mode we employ such techniques as bagging, subagging and repeated cross-validation. Theoretical analyses and experimental results to date proved benefits of

bagging especially in terms of stability improvement and variance reduction of learners for both classification and regression problems [3], [5-8], [13], [14].

2 Multi-Agent System for Real Estate Appraisal

We started developing the second version of a multi-agent system to aid in real estate appraisal, which could be deployed and used in cadastral information centres in Poland. So, we decided to program it using Spring Framework, which is one of the most popular solutions for Java web application development. Spring Framework is an open source application framework which supports mechanisms like *dependency injection* (DI) and *inversion of control* (IoC) making code testable and much easier to maintain. Moreover the Spring provides a set of tools like: the Spring ORM - support for object relational mapping, the Spring DAO - transaction infrastructure and DAO support, the Spring MVC - *model-view-controller* paradigm implementation. From our point of view the Spring Integration turns out to be the most interesting module.

The Spring Integration is an implementation of *Enterprise Integration Patterns* presented by Hohpe and Woolf in [10]. It helps to build an enterprise integration layer aside of application core by providing message driven approach. Situation in which all communication problems are handled by the Spring Integration gives us ability to focus on key multi-agent system features. The Spring Integration communication model is based on several main concepts. A *Message* is a generic wrapper for any Java object we want to send. It consists a message header and payload. All communication is performed through *Message Channels*. The last important element is the *Message Endpoint*. The *Message Endpoint* (producer and consumer) constitute glue between message channels which belongs to infrastructure layer and application domain.

The general architecture of the MAREA-2 system is presented in Figure 2 and it comprises following agents:

Interface Agents. Interface agents interact with appraisers, i.e. system users, gather parameters of properties being appraised and present suggested prices of properties exposed to valuation.

Query Agents. Query agents are generated at each demand of a user. They apply to available models assisting with real estate appraisals. These models are used to generate the suggested prices of properties being appraised.

Result Agents. Result agents process the results provided by the models assisting with valuation and prepare suggested prices to be presented to the users.

Data Agents. Data agents process data gathered in source databases. They extract, cleanse and integrate data preparing them to be useful to learn real estate appraisal models. They create and maintain a certain kind of appraisal data warehouse. Moreover they accomplish various data splits and feature selection adequate to selected machine learning methods.

Learning Agents. Learning agents perform overall and supplementary learning, creating in result different data driven models for real estate appraisal.

Aggregating Agents. Aggregating agents select generated models and create ensembles according to user queries.

Source databases are heterogeneous and distributed, they are composed of a cadastral database, cadastral digital map and registry of real estate sales/purchase transactions. The source databases are continuously updated by an information centre located at a self-government office. *Data Agents* retrieve data necessary to build appraisal models, cleanse, integrate them, and finally save into an appraisal data warehouse. *Learning Agents* create data driven appraisal models from scratch or update existing ones based on the analysis of data included in the warehouse. Remaining agents serve the appraisers: *Interface Agent* communicates with a user and sends his request to *Query Agent*, which in turn, formulates a query and passes it to *Aggregating Agent*. Next, *Aggregating Agent* selects an optimal subset of appraisal models and points it out to *Result Agent*. Then, *Result Agent* generates the final response and passes it back to *Interface Agent*.

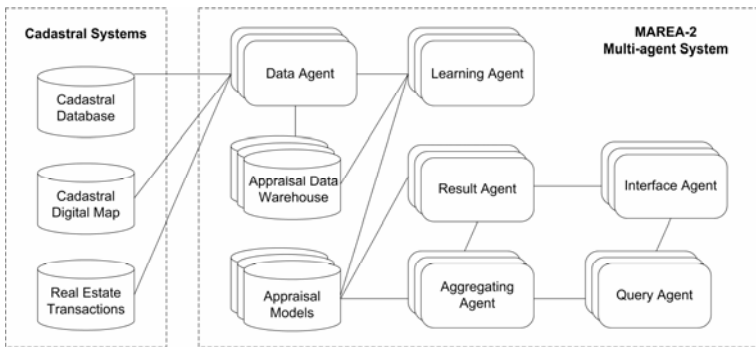


Fig. 2. General architecture of the MAREA-2 system

3 Description of Experiments

The comparative investigation of bagging, subbagging and repeated cross-validation resampling techniques was conducted using six JAVA learner classes available in WEKA library [15]:

LR – Linear Regression. Algorithm is a standard statistical approach to build a linear model predicting a value of the variable while knowing the values of the other variables. It uses the least mean square method in order to adjust the parameters of the linear model.

M5P – Pruned Model Tree. Implements routines for generating M5 model trees. The algorithm is based on decision trees, however, instead of having values at tree's nodes, it contains a multivariate linear regression model at each node. The input space is divided into cells using training data and their outcomes, then a regression model is built in each cell as a leaf of the tree.

M5P Rules – Pruned Model Tree. The algorithm divides the parameter space into areas (subspaces) and builds in each of them a linear regression model. It is based on M5 algorithm. In each iteration a M5 Tree is generated and its best rule is extracted according to a given heuristic.

SMOreg – *Support Vector Machine for Regression*. Finding a maximum margin hyperplane applies only to classification problems. For prediction problems model needs to be trained with help of *Sequential Minimal Optimization* algorithm (*SMO*). In our case we used polynomial kernel for that purpose.

MLP – *Multi Layer Perceptron*. One of the most popular neural network. It uses backpropagation for training. In our experiment we used one hidden layer. In output layer there was only one neuron presenting prediction result.

RBF Network – *Radial Basis Function Network*. It consist of input layer, one hidden layer with radial basis function as activation function and output layer. Start positions of basis function centers is found using k-means clustering algorithm choosing cluster means as the centers.

In order to compare machine learning algorithms with techniques applied to property valuation we asked experts to evaluate premises using their pairwise comparison method to historical data of sales/purchase transactions recorded in a cadastral system. The experts worked out a computer program which simulated their routine work and was able to estimate the experts’ prices of a great number of premises automatically. They selected five following features as main drivers of premises prices: usable area of premises, age of a building, number of rooms in a flat, number of storeys in a building, and distance from the city centre.

Real-world dataset used in experiments was drawn from a rough dataset containing above 50 000 records referring to residential premises transactions accomplished in one Polish big city with the population of 640 000 within 11 years from 1998 to 2008. The final dataset counted the 5213 samples for which the experts could estimate the value using their pairwise comparison method. Due to the fact that the prices of premises change substantially in the course of time, the whole 11-year dataset cannot be used to create data-driven models, therefore it was split into 20 half-year subsets. The sizes of half-year data subsets are given in Table 1.

Table 1. Number of instances in half-year datasets

1998-2	1999-1	1999-2	2000-1	2000-2	2001-1	2001-2	2002-1	2002-2	2003-1
202	213	264	162	167	228	235	267	263	267
2003-2	2004-1	2004-2	2005-1	2005-2	2006-1	2006-2	2007-1	2007-2	2008-1
386	278	268	244	336	300	377	289	286	181

Following resampling methods and their variants were applied in experiments and compared with the standard 10cv and the experts’ method.

Bagging: *B100, B70, B50, B30* – m-out-of-n bagging with replacement with different sizes of samples using the whole base dataset as a test set. The numbers in the codes indicate what percentage of the base set was drawn to create training sets.

Subagging: *S90, S70, S50, S30* – m-out-of-n bagging without replacement (i.e. repeated holdout) with different sizes of samples. The numbers in the codes point out what percentage of the base dataset was drawn to create a training set.

Repeated cross-validation: *2x50cv, 5x20cv, 10x10cv, 20x5cv, 50x2cv* – k-fold cross-validation splits, for $k=50, 20, 10, 5,$ and 2 respectively, were repeated 2, 5, 10, 20, and 50 times respectively, to obtain 100 pairs of training and test sets.

In the case of bagging and subbagging methods 100 replicates (bags) were created on the basis of each base dataset, as performance functions the mean square error (MSE) was used, and as aggregation functions simple averages were employed. The normalization of data was accomplished using the min-max approach.

4 Results of Experiments

The performance of *Bagging*, *Subbagging*, and *Repeated cross-validation* models created by six machine learning algorithms was statistically analysed and compared with MSE provided by *10cv*, applied usually to evaluate single models, and *Expert* methods. The Friedman test done in respect of MSE values of all models built over 20 one-year datasets indicated significant differences between some models. Average ranks of individual models for respective resampling techniques are shown in Tables 2, 3, and 4, where the lower rank value the better model. In Tables 5-10 the results of non-parametric Wilcoxon signed-rank test to pairwise comparison of the model performance are presented. The zero hypothesis stated there were not significant differences in accuracy, in terms of MSE, between given pairs of models. In both tables + denotes that the model in the row performed significantly better than, - significantly worse than, and ≈ statistically equivalent to the one in the corresponding column, respectively. In turn, / (slashes) separate the results for individual methods. The significance level considered for the null hypothesis rejection was 5%.

Table 2. Average rank positions of bagging models determined during Friedman test

Method	1st	2nd	3rd	4th	5th	6th
LR	B100 (1.95)	B70 (2.45)	B50 (3.40)	10cv (3.80)	B30 (4.45)	Expert (4.95)
M5P	B100 (1.00)	B70 (2.05)	B50 (3.15)	B30 (4.30)	10cv (4.95)	Expert (5.55)
M5R	B100 (1.00)	B70 (2.05)	B50 (3.05)	B30 (4.10)	10cv (5.30)	Expert (5.50)
SMO	10cv (2.60)	B70 (3.30)	B100 (3.35)	B50 (3.60)	B30 (3.65)	Expert (4.50)
MLP	B100 (2.10)	B70 (2.15)	B50 (2.35)	B30 (3.40)	Expert (5.45)	10cv (5.55)
RBF	Expert (1.95)	B30 (2.50)	B50 (3.20)	B100 (3.80)	B70 (4.20)	10cv (5.35)

Table 3. Average rank positions of subbagging models determined during Friedman test

MetSod	1st	2nd	3rd	4tS	5tS	6tS
LR	10cv (1.55)	S90 (2.45)	S70 (2.95)	S50 (4.20)	Expert (4.40)	S30 (5.45)
M5P	10cv (1.85)	S90 (2.10)	S70 (2.95)	Expert (4.30)	S50 (4.35)	S30 (5.45)
M5R	10cv (1.90)	S90 (2.15)	S70 (3.15)	Expert (3.85)	S50 (4.35)	S30 (5.60)
SMO	10cv (1.65)	S90 (2.65)	S70 (2.85)	S50 (4.25)	Expert (4.40)	S30 (5.20)
MLP	S50 (1.90)	Expert (2.15)	10cv (3.15)	S90 (3.85)	S70 (4.35)	S30 (5.60)
RBF	Expert (1.65)	S90 (2.90)	10cv (3.45)	S70 (3.95)	S50 (4.35)	S30 (5.60)

Table 4. Average rank positions of cross-validation models determined by Friedman test

Method	1st	2nd	3rd	4th	5th	6th
LR	2x50cv (1.05)	5x20cv (2.10)	10x10cv (3.20)	20x5cv (4.30)	Expert (5.05)	50x2cv (5.30)
M5P	2x50cv (1.00)	5x20cv (2.10)	10x10cv (3.15)	20x5cv (4.30)	Expert (5.10)	50x2cv (5.30)
M5R	2x50cv (1.00)	5x20cv (2.20)	10x10cv (3.35)	20x5cv (4.20)	Expert (4.80)	50x2cv (5.45)
SMO	2x50cv (1.05)	5x20cv (2.10)	10x10cv (3.30)	20x5cv (4.30)	Expert (4.95)	50x2cv (5.30)
MLP	2x50cv (1.55)	Expert (2.60)	5x20cv (2.80)	10x10cv (3.70)	20x5cv (4.65)	50x2cv (5.70)
RBF	2x50cv (1.75)	Expert (1.80)	5x20cv (2.80)	10x10cv (4.00)	20x5cv (5.25)	50x2cv (5.40)

Table 5. Results of Wilcoxon tests for bagging models for LR/M5P/M5R

	B100	B70	B50	B30	10cv	Expert
B100		+/+/+	+/+/+	+/+/+	≈/+/+	+/+/+
B70	-/-/-		+/+/+	+/+/+	≈/+/+	+/+/+
B50	-/-/-	-/-/-		+/+/+	≈/+/+	+/+/+
B30	-/-/-	-/-/-	-/-/-		≈/+/+	+/+/+
10cv	≈/-/-	≈/-/-	≈/-/-	≈/-/-		+/+/+
Expert	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	

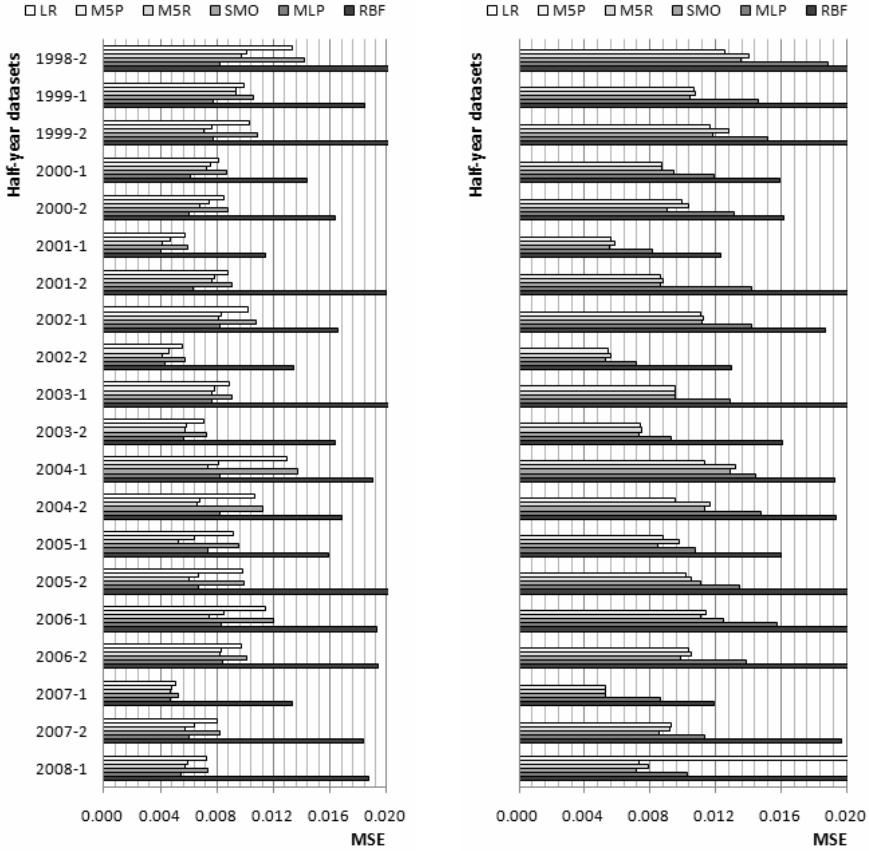


Fig. 3. Performance of *B100* (left) and *S90* (right) models

Table 6. Results of Wilcoxon tests for bagging models for SMO/MLP/RBF

	B100	B70	B50	B30	10cv	Expert
B100		≈/≈/+	≈/≈/+	≈/+/-	-/+/+	+/+/-
B70	≈/≈/-		≈/≈/-	≈/+/-	-/+/+	+/+/-
B50	≈/≈/-	≈/≈/+		≈/+/-	-/+/+	+/+/-
B30	≈/-/+	≈/-/+	≈/-/+		-/+/+	+/+/-
10cv	+/-/-	+/-/-	+/-/-	+/-/-		+/≈/-
Expert	-/-/+	-/-/+	-/-/+	-/-/+	-/≈/+	

Table 7. Results of Wilcoxon tests for subgging models for LR/M5P/M5R

	S90	S70	S50	S30	10cv	Expert
S90		+ / + / +	+ / + / +	+ / + / +	- / ≈ / ≈	+ / + / +
S70	- / - / -		+ / + / +	+ / + / +	- / - / -	+ / + / ≈
S50	- / - / -	≈ / - / -		+ / + / +	- / - / -	≈ / ≈ / ≈
S30	- / - / -	- / - / -	- / - / -		- / - / -	≈ / ≈ / ≈
10cv	+ / ≈ / ≈	+ / + / +	+ / + / +	+ / + / +		+ / + / +
Expert	- / - / -	- / - / ≈	≈ / ≈ / ≈	≈ / ≈ / ≈	- / - / -	

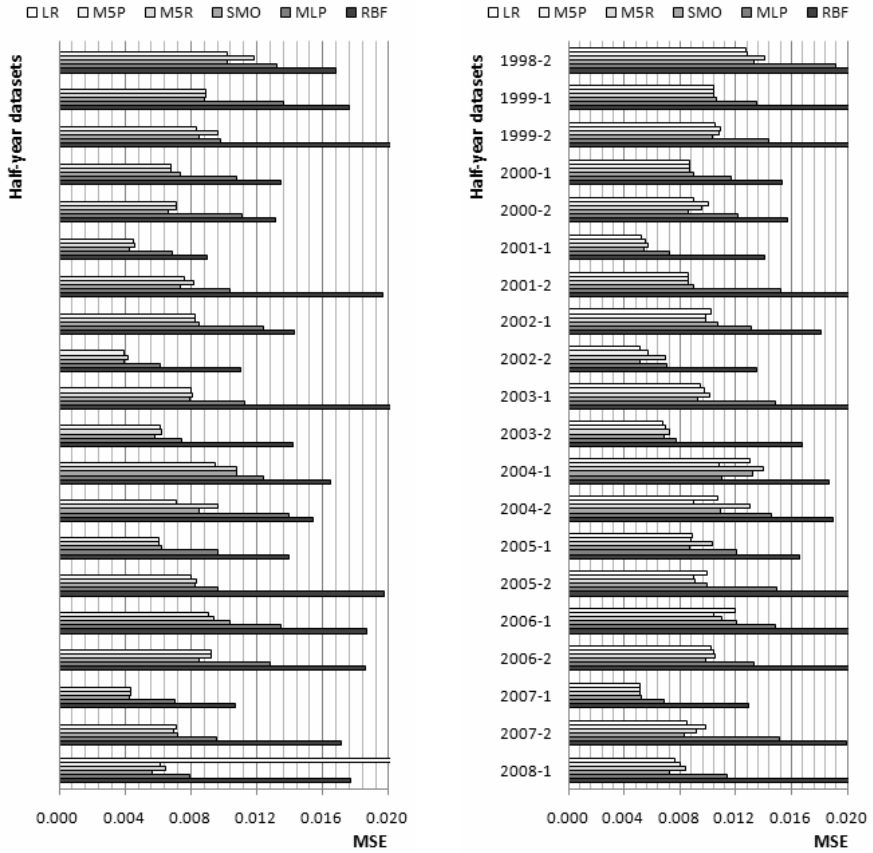


Fig. 4. Performance of 2x50cv (left) and 10cv (right) models

Table 8. Results of Wilcoxon tests for subgging models for SMO/MLP/RBF

	S90	S70	S50	S30	10cv	Expert
S90		≈ / + / ≈	+ / - / +	+ / + / +	- / ≈ / ≈	+ / ≈ / -
S70	≈ / - / ≈		+ / - / ≈	+ / + / +	- / - / -	+ / ≈ / -
S50	- / + / -	- / + / ≈		+ / + / +	- / + / -	≈ / ≈ / -
S30	- / - / -	- / - / -	- / - / -		- / - / -	≈ / - / -
10cv	+ / ≈ / ≈	+ / + / +	+ / - / +	+ / + / +		+ / ≈ / -
Expert	- / ≈ / +	- / ≈ / +	≈ / ≈ / +	≈ / + / +	- / ≈ / +	

Table 9. Results of Wilcoxon tests for cross-validation models for LR/M5P/M5R

	2x50cv	5x20cv	10x10cv	20x5cv	50x2cv	Expert
2x50cv		+ / + / +	+ / + / +	+ / + / +	+ / + / +	+ / + / +
5x20cv	- / - / -		+ / + / +	+ / + / +	+ / + / +	+ / + / +
10x10cv	- / - / -	- / - / -		+ / + / +	+ / + / +	+ / + / +
20x5cv	- / - / -	- / - / -	- / - / -		+ / + / +	+ / + / +
50x2cv	- / - / -	- / - / -	- / - / -	- / - / -		≈ / ≈ / ≈
Expert	- / - / -	- / - / -	- / - / -	- / - / -	≈ / ≈ / ≈	

Table 10. Results of Wilcoxon tests for cross-validation models for SMO/MLP/RBF

	2x50cv	5x20cv	10x10cv	20x5cv	50x2cv	Expert
2x50cv		+ / + / +	+ / + / +	+ / + / +	+ / + / +	+ / ≈ / +
5x20cv	- / - / -		+ / + / +	+ / + / +	+ / + / +	+ / ≈ / -
10x10cv	- / - / -	- / - / -		+ / + / +	+ / + / +	+ / ≈ / -
20x5cv	- / - / -	- / - / -	- / - / -		+ / + / ≈	+ / ≈ / -
50x2cv	- / - / -	- / - / -	- / - / -	- / - / ≈		≈ / ≈ / -
Expert	- / ≈ / -	- / ≈ / +	- / ≈ / +	- / ≈ / +	≈ / ≈ / +	

Table 11. Average rank positions of selected models determined by Friedman test

Method	1st	2nd	3rd	4th	5th	6th
B100	M5R (1.45)	MLP (1.85)	M5P (2.70)	LR (4.00)	SMO (5.00)	RBF (6.00)
S90	LR (1.85)	M5P (2.23)	SMO (2.45)	M5R (3.48)	MLP (5.00)	RBF (6.00)
2x50cv	SMO (2.10)	M5P (2.18)	LR (2.50)	M5R (3.48)	MLP (5.00)	RBF (6.00)
10cv	LR (2.00)	M5P (2.45)	SMO (2.50)	M5R (3.20)	MLP (4.85)	RBF (6.00)

The results obtained for the best models of individual types of resampling techniques, i.e. B100, S90, 2x50cv and for single models (10cv) are illustrated in Fig. 3 and 4 and their rank positions determined by Friedman test are placed in Table 11.

The general outcome is as follows: the performance of the experts' method fluctuated strongly achieving for some datasets excessively high MSE values and for others the lowest values; MSE ranged from 0.007 to 0.023. Majority of models, but one (RBF), provided better or equivalent accuracy than the experts' method. The models created over 30% subsamples performed significantly worse than ones trained using bigger portions of base datasets for all methods. The RBF revealed the worst performance so that it should be discarded. M5P, M5R, and MLP bagging ensembles outperformed their corresponding single models. SMO and LR bagging ensembles gave no gain in accuracy compared to their corresponding single models. For subbagging the definite majority of ensembles performed worse than their single counterparts. In turn, for repeated cross-validation the bigger size of training set the better accuracy of models. Moreover, 10x10cv produced statistically equivalent results as 10cv, therefore only the former was shown in respective tables and figures.

5 Conclusions and Future Work

A new version of the a multi-agent system to aid in real estate appraisal, called MAREA-2, was introduced. The system is being developed using Java Spring Framework and is intended for industrial application in cadastral information centres in Poland maintaining cadastral systems with digital cadastral maps and property sales

transaction registries. The core of the system are appraisal models which are created and updated by means of machine learning algorithms.

The major part of the study was devoted to investigate the performance of *Bagging*, *Subbagging*, and *Repeated cross-validation* models created by six machine learning algorithms compared to MSE provided by 10cv, applied usually to evaluate single models, and Expert methods.

The overall result of our investigation was that the majority of models created using resampling techniques provided better or equivalent accuracy than the experts' method employed in reality. It confirms that automated valuation models can be successfully incorporated into the multi-agent system and be utilized to support appraisers' work.

We plan to continue developing the MAREA-2 system and exploring resampling methods ensuring faster data processing such as random subspaces, subsampling, and techniques of determining the optimal sizes of multi-model solutions. This can lead to achieve both low prediction error and an appropriate balance between accuracy, complexity, stability.

Acknowledgments. This paper was partially supported by Grant no. N N519 444939 funded by Polish Ministry of Science and Higher Education (2010-2013).

References

1. Bańczyk, K.: Multi-agent system based on heterogeneous ensemble machine learning models. Master's Thesis, Wrocław University of Technology, Wrocław, Poland (2011)
2. Biau, G., Cérou, F., Guyader, A.: On the Rate of Convergence of the Bagged Nearest Neighbor Estimate. *Journal of Machine Learning Research* 11, 687–712 (2010)
3. Borra, S., Di Ciaccio, A.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis* 54(12), 2976–2989 (2010)
4. Breiman, L.: Bagging Predictors. *Machine Learning* 24(2), 123–140 (1996)
5. Bühlmann, P., Yu, B.: Analyzing bagging. *Annals of Statistics* 30, 927–961 (2002)
6. Burman, P.: A comparative study of ordinary cross-validation, v -fold cross-validation and the repeated learning-testing methods. *Biometrika* 76, 503–514 (1989)
7. Friedman, J.H., Hall, P.: On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference* 137(3), 669–683 (2007)
8. Fumera, G., Roli, F., Serrau, A.: A theoretical analysis of bagging as a linear combination of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(7), 1293–1299 (2008)
9. Graczyk, M., Lasota, T., Telec, Z., Trawiński, B.: A Multi-Agent System to Assist with Property Valuation Using Heterogeneous Ensembles of Fuzzy Models. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6070, pp. 420–429. Springer, Heidelberg (2010)
10. Hohpe, G., Woolf, B.: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, Boston (2004)
11. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Concept of a Multi-Agent System for Assisting in Real Estate Appraisals. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009*. LNCS, vol. 5559, pp. 50–59. Springer, Heidelberg (2009)

12. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: A Multi-agent System to Assist with Real Estate Appraisals Using Bagging Ensembles. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 813–824. Springer, Heidelberg (2009)
13. Molinaro, A.N., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. *Bioinformatics* 21(15), 3301–3307 (2005)
14. Rodriguez, J.D., Perez, A., Lozano, J.A.: Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(3), 569–575 (2010)
15. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Voting in Multi-Agent System for Improvement of Partial Observations

Dariusz Choinski, Mieczyslaw Metzger, and Witold Nocon

Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{dariusz.choinski,mieczyslaw.metzger,witold.nocon}@polsl.pl

Abstract. In this paper a system for monitoring a biotechnical system is presented. Some of the events associated with the state of metabolic reactions are indistinguishable, mainly due to lack of appropriate sensors and measurement capabilities. Therefore, a solution is needed to identify the state in which the reactor currently is, based on the information and measurements available in real time. The solution presented in this paper is based on a multi agent system, in which particular agents identify the state of the process based on selected measurements. Those partial identification results are then used to provide a cumulative result by means of a voting mechanism between all the particular agents. Such a solution seems to be a promising alternative to standard monitoring and identification methods.

Keywords: Voting, Multi-Agent Systems, Monitoring of Biotechnological Processes.

1 Introduction

In many industrial processes, especially in the field of biotechnology, classical control algorithms are not capable of providing a satisfactory maintenance of the process. This is because such processes are difficult to fully observe and monitor in real time. Despite having better and better measurement instruments, it is still not enough to gather all the data needed in order to find the best control strategies. Using agent technology in such processes [1], [2], together with reinforcement learning and voting mechanisms, seems to be a promising alternative. Reinforcement learning [3] is a method describing how to use some events and situations based on trial-and-error methods of generating actions so that the numerical information is fully used. A Q-learning method [4], [5] is related to the reinforcement learning. It has been proposed for discrete systems [6] and then expanded to continuous state systems [7]. A theoretical description of those techniques may be found in [8], [9], while theoretical description of voting in agent systems may be found in [10], [11]. Conitzer [12] in his review paper discusses multi agent systems in combinatorial auctions and voting.

In manufacturing control and discrete systems the employment of MAS together with Q-learning may be applied in applications such as distributed reinforcement learning control for batch sequencing and sizing in just-in-time manufacturing systems [13], cooperation of multirobots [14], multi agent cooperation for robot soccer [15], application of visual information in target reaching [16], six-legged walking

machine [17], supplied optimal bidding strategy in electricity pay-as-bid auction [18] and even multi-agent learning for routing control within an Internet environment [19].

Despite the fact that agent systems related to Q-learning are not employed in automation of continuous process very often, some applications may still be found. In [20] application of those methods have been presented for pH control, while in [21] an autonomous agent that continuously learns to implement control actions that can drive the batch process state very close to the desired one with near-optimal performance have been presented. Comparing the quality of solutions in transportation problems has been presented in [22]. An algorithm for reinforcement learning called fuzzy Q-Learning is proposed to build the agent (controller). A small amount of applications in control of continuous processes is increased in this work, in which a multi agent reinforcement learning is employed for a biotechnological pilot plant.

The paper is organized as follows. After this introduction, in section 2, the problem of voting in multi agent system for improvement of partial observation is formulated and the specific solution that relates to the biotechnological pilot plant is outlined. In section 3, the biotechnological plant is presented together with the architecture of the control system realizing plant level and supervisory level control. In section 4, the implementation of the multi agent system under investigation is presented, while section 5 presents some concluding remarks.

2 Problem Formulation

Supervisory control within a hybrid system designed for the purpose of a biotechnological pilot-plant control [23], [24] is based on observation of sequences of events Φ generated by the control layer. Those events enable the state of the reactor to be identified. The reactor may be in one of the following states: primary growth – growth of biomass mainly due to utilization of substrate being taken up by the biomass; secondary growth – growth of biomass due to utilization of substrate previously accumulated; aggregation and hydrolysis of extracellular polymers [25]. To some extent, those states represent feast and famine conditions, since in the feast conditions, substrate take up is being used for growth and is also accumulated by the biomass (in a form of PHB – polyhydroxybutyrate), while in famine conditions, PHB is being utilized for growth and maintenance of biomass since primary substrate is not available. Identification of those states is important from the point of view of control, since different control strategies should be employed by the supervisory level control system in different bioreactor states.

Some of the events associated with the state of metabolic reactions are indistinguishable, mainly due to lack of appropriate sensors and measurement capabilities. The biological reactor is equipped with standard, off-the-shelf sensors. Those measurements are: dissolved oxygen concentration [mg O₂/l], off-gas carbon dioxide [ppm] and oxidation-reduction potential [mV] which indicates the balance between reduction and oxidation reactions predominant in the reactor. Additionally, pH value is also measured. The technically measurable values provide information that is in reality dependent on many chemical and biological phenomena. Those phenomena reflect the state of metabolism and depending on the current reactor conditions, only a subset of those measurements enable a correct observation of the system. The

problem is additionally complicated because it is necessary to differentiate the measurements on-line. Only the characteristic points in a differentiated signals enable the events taking place in the reactor to be identified. The aeration system used to maintain a desired concentration of dissolved oxygen in the reactor causes additional disturbance. Aeration influences all other measurements. For example, aeration also causes the off-gas carbon dioxide concentration reading to jump, although this increase is not due to biological reaction but is caused by an increase of mass gas flow through the reactor. Fortunately, is it known when different controls are engaged, hence it is possible to select a proper function describing the state of the process and take the aeration process into account. Additionally, the aeration process is indicated in all available measurements.

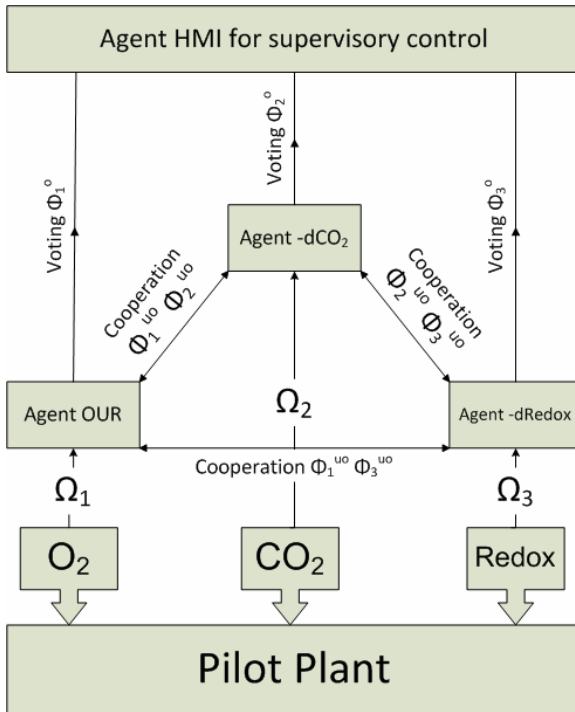


Fig. 1. Idea of voting in Multi-Agent System for improvement in partial observations

Because of the described disturbances, the algorithms implemented in the software Agents: OUR, -dCO₂ and -Redox, may independently generate events based on the measurements representing different state variables Ω (Fig. 1). Those events may be divided into observable events Φ^o and unobservable events Φ^{ou} . This results in a partial observation of the system by the supervisory control layer of the Distributed Control System [26]. The division of events has been introduced based on [26] and relates to a pure physical situation, in which some of the events are redirected within the

MAS into the expert control layer. This redirection is caused by constraints of measurement devices. In this layer agents using uncertain data may be excluded. Fragments of the observed measurements that relate to the unobservable events resulting from measurement disturbances and aeration control are checked by cooperation with other agents. However, such an algorithm may result in different results depending on the history of different agents and random sequence of information passed within cooperation. Therefore, in the cooperation of agents, only information reading is performed without passing the conclusions derived by different agents. In every agent, the information gathered from other agents is used for establishing the correlation of events that concur in all measurements. Particular agents are not informed about the results obtained by other Agents. Instead, each agent assesses its own results in observable event identification using a previously agreed upon system of scoring. This scoring together with a set of observable events regarding the current state of the plant is passed to HMI Agent. This agent is responsible for selecting results based on the majority of votes. Those results, together with previously agreed upon weight factors, serve as a basis for plant evaluation on the supervisory control level [27]. Such an architecture was also evaluated and presented in [28]. Scoring ranges and weight factors are periodically computed based on laboratory measurements that assess the multi-agent system at the same time.

3 Biotechnological Pilot Plant under Investigation

The biotechnological process pilot-plant [29] designed and operated at the Faculty of Automatic Control, Electronics and Computer Science serves as a platform for investigations regarding activated sludge process in aquatic environment (Fig. 2). Apart from the standard activated sludge process, bioaugmentation process may also be investigated. The pilot-plant consists of a biological reactor and a settler with a number of controls (pumps, mixers, heater) and measurements, together with a programmable logic controller capable of running advanced control algorithms (based on control systems of different vendors).

Depending on the particular set of control algorithms applied, the structure of the biological process involved may be changed. For example, the plant may be operated as a continuous or sequencing activated sludge process, the later involving cyclic utilization of the biological reactor for reaction and settling phases, the former involving continuous sedimentation of activated sludge in the settler with recycle of the thickened sludge back to the reactor. Depending on the control of pumps interchanging the sludge between the reactor and the settler, the biological reactor may either be operated as a chemostat (constant reaction rate in the reactor) or as a turbidostat (constant concentration of biomass in the reactor). Investigations regarding batch processes may require a predefined initial concentration of biomass in the reactor, therefore the settler should be used as a buffer for the sludge and the reactor should be operated with changing level. And last but not least, bioaugmentation of biomass in the plant may or may not be involved.

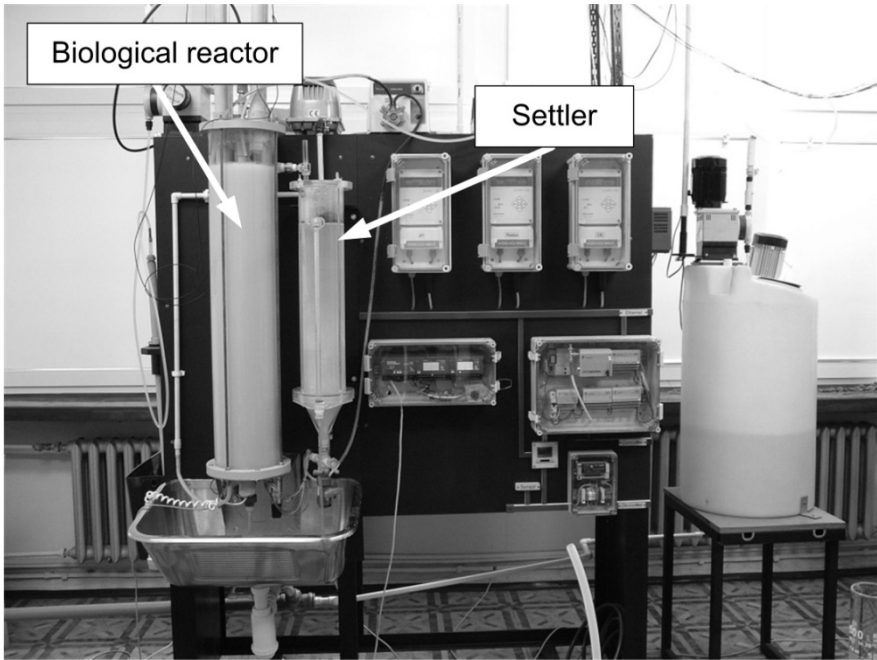


Fig. 2. Biotechnological pilot plant operated at the Faculty of Automatic Control, Electronics and Computer Science

Due to a biological nature of the process, the pilot plant requires an uninterrupted control 24 hours a day, 7 day a week. If, for any reason, the biomass inside the system is not controlled properly, the quality and properties of the activated sludge will change drastically, thus preventing the researchers from performing honest experimental studies. Therefore, a fault-tolerant control system is necessary. The control software responsible for supervisory operation of the plant is run on a Windows based PC computer and is implemented using the LabVIEW programming language. Such a solution ensured that implementation and modification of advanced algorithms is easy and straightforward. A PC-based solution however is the most error prone element in the system. Therefore, a National Instruments FP-2010 controller provides a real-time system that runs a software, written in LabVIEW also, realizing the plant level control algorithms. In case of a failure of the supervisory control layer, the plant-level control layer may take over part of the supervisory level functions.

In order to enable connection of the existing distributed control system (DCS) to industry grade programmable logic controllers (PLC) and/or to industry grade supervisory control and data acquisition (SCADA) system, measurements and controls are presented using a National Instruments Shared Variable Engine (SVE) [30]. The SVE is responsible amongst other things for serving up data through the OPC thus enabling data to be available by other software.

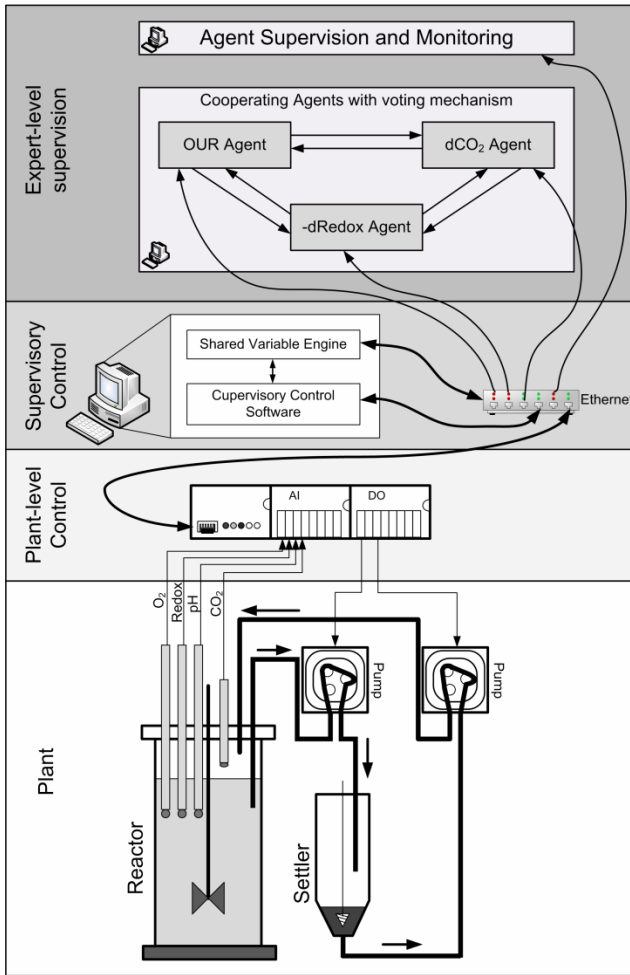


Fig. 3. Architecture of the Multi agent system controlling the biological pilot plant

Fig. 3 presents a scheme of the Multi-Agent System (MAS) build upon the supervisory control system. The MAS is accessing data (measurements and controls) published by the SVE using Ethernet network. The MAS is also capable of writing data to the SVE, thus enabling particular agents to take over control of certain parts of the process. In such a case, special attention must be paid when dealing with control of dynamical processes requiring uninterrupted control [31].

The lowest level of control – the plant level control – basic control algorithms are implemented that ensure a safe operation of the biological system under investigation. Those control algorithms are:

- algorithms for controlling pumps in the system in order to ensure a proper mass flow between the reactor and the settler and proper substrate additions,

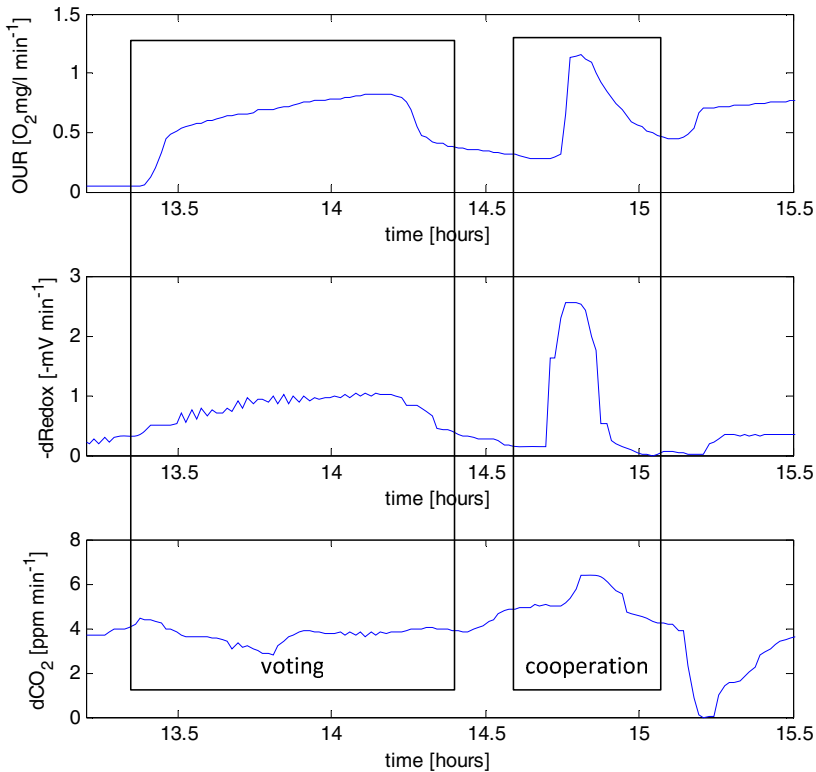


Fig. 4. Example of experimental results

- algorithms for control of mixers in the reactor and in the settler. The purpose of those mixers is to ensure a decantation of the reactor to be performed (by periodically stopping the mixer in the reactor thus enabling the clarified liquid to be removed) and also to enable mass balance control (by controlling retention of biomass in the settler by increased or decreased mixing in the settler),
- algorithms for control of dissolved oxygen concentration in the reactor by means of aeration, thus enabling control of either nitrification or denitrification processes in the reactor.

Those low-level control algorithms are able to maintain a proper operation of the biological system in a time horizon of hour to a couple of days. In a longer time horizon, a supervisory control system is needed in order to monitor trends in measured variables, react to different composition of wastewater (different substrate) and to define sequences of set point variables.

4 Implementation of Expert Control for the Biotechnical Process

The top most layer in the control system presented in Fig. 3 is the expert level. It is realized as a MAS, since three independent software agents are cooperating in order

to analyze the available measurements and thus identifying the current microbiological state in which the biomass in the reactor currently is. Those states may be generally described as:

- accumulation of substrate (A),
- utilization of substrate for growth and endogenous processes (U),
- constant substrate addition (C).

Particular states may be active at the same time, hence the current state of the reactor may be described as one of the following:

- A/U/C – concurrent accumulation, utilization and constant loading,
- A/U – concurrent accumulation and utilization,
- A – accumulation only,
- C – constant loading of substrate only.

In order to identify in which state the reactor currently is, three different software agents have been realized, each using different measurements. Those agents are:

- -dRedox Agent – using the Redox potential for identification,
- dCO₂ Agent – using carbon dioxide for identification,
- OUR Agent – using the measured dissolved oxygen concentration measurement for identification. Based on this measurement, the OUR Agent computes the oxygen uptake rate (OUR) in order to determine the current load of substrate on the system.

Each agent autonomously analyzes the measurement data available, and based on those measurements identifies the current state of the reactor. Analysis based only on selected measurements may not be credible. Therefore, each agent possesses information about states identified by other agents. Therefore, agents perform voting on the state in which the majority of agents agrees upon. In the particular case of three cooperating agents, the voting itself is rather straightforward. The collective analysis of reactor state results in the most probable state to be presented as the result of identification. Fig. 4 presents an example of experimental results.

A forth agent designed for supervision and monitoring is provided to enable easy monitoring of the identification processes performed by other agents.

5 Concluding Remarks

In this paper a system for monitoring a biotechnical system has been presented. The basis for this system is a multi agent system in which three agents try to identify the state of the process using a different set of measurements. The voting mechanism, allows the best agreed upon solution to be proposed as a solution to the overall identification problem. Because, the supervisory layer presents data through OPC, it may be integrated with existing industry grade control systems.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education, grant no. N N514 471539 and by the Silesian University of Technology BW/RGH-2/RAu0/2010.

References

1. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. In: *Autonomous Agents and Multi-Agent Systems*, vol. 1, pp. 7–38. Kluwer Academic Publishers, Boston (1998)
2. Weiss, G. (ed.): *MultiAgent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge (1999)
3. Sutton, R.S., Barto, A.G.: *Reinforcement learning - an introduction*. MIT Press, Cambridge (1998)
4. Watkins, C.J.C.H.: *Learning from delayed rewards*. Ph.D. thesis. University of Cambridge, England (1989)
5. Watkins, C.J.C.H., Dayan, P.: Technical note: Q-learning. *Machine Learning* 8, 279–292 (1992)
6. Cassandra, C.G., Lafortune, S.: *Introduction to discrete event system*. Springer, New York (2008)
7. Werbos, P.J.: Approximate dynamic programming for real-time control and neural modeling. In: *Handbook of Intelligent Control*. Van Nostrand Reinhold, New York (1992)
8. Kim, J.-H., Lewis, F.L.: Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI. *Automatica* 46, 1320–1326 (2010)
9. Murakoshi, K., Mizuno, J.: A parameter control method in reinforcement learning to rapidly follow unexpected environmental changes. *BioSystems* 77, 109–117 (2004)
10. Croll, P.R., Sharkey, A.J.C., Bass, J.M., Sharkey, N.E., Fleming, P.J.: Dependable, Intelligent Voting for Real-time Control Software. *Engng. Applic. Artif. Intell.* 8, 615–623 (1995)
11. Latif-Shabgahi, G., Bennet, S., Bass, J.M.: Smoothing voter: a novel voting algorithm for handling multiple errors in fault-tolerant control systems. *Microprocessors and Microsystems* 27, 303–313 (2003)
12. Conitzer, V.: Comparing multiagent systems research in combinatorial auctions and voting. *Ann. Math. Artif. Intell.* 58, 239–259 (2010)
13. Hong, J., Prabhu, V.V.: Distributed Reinforcement Learning Control for Batch Sequencing and Sizing in Just-In-Time Manufacturing Systems. *Applied Intelligence* 20, 71–87 (2004)
14. Ohshita, T., Shin, J.-S., Miyazaki, M., Lee, H.-H.: A cooperative behavior learning control of multi-robot using trace information. *Artif. Life Robotics* 13, 144–147 (2008)
15. Park, K.-H., Kim, Y.-J., Kim, J.-H.: Modular Q-learning based multi-agent cooperation for robot soccer. *Robotics and Autonomous Systems* 35, 109–122 (2001)
16. Distanto, C., Anglani, A., Taurisano, F.: Target Reaching by Using Visual Information and Q-learning Controllers. *Autonomous Robots* 9, 41–50 (2000)
17. Kirchner, F.: Q-learning of complex behaviours on a six-legged walking machine. *Robotics and Autonomous Systems* 25, 253–262 (1998)
18. Rahimiyan, M., Mashhadi, M.R.: Supplier's optimal bidding strategy in electricity pay-as-bid auction: Comparison of the Q-learning and a model-based approach. *Electric Power Systems Research* 78, 165–175 (2008)
19. Tillotson, P.R.J., Wu, Q.H., Hughes, P.M.: Multi-agent learning for routing control within an Internet environment. *Engineering Applications of Artificial Intelligence* 17, 179–185 (2004)
20. Syafie, S., Tadeo, F., Martinez, E.: Model-free learning control of neutralization processes using reinforcement learning. *Engineering Applications of Artificial Intelligence* 20, 767–782 (2007)
21. Wilson, J.A., Martinez, E.C.: Neuro-fuzzy modeling and control of a batch process involving simultaneous reaction and distillation. *Computers Chem. Engng.* 21, S1233–S1238 (1997)

22. Momot, A., Małysiak-Mrozek, B.z., Kozielski, S., Mrozek, D., Hera, Ł., Górczyńska-Kosiorz, S., Momot, M.: Improving Performance of Protein Structure Similarity Searching by Distributing Computations in Hierarchical Multi-Agent System. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS, vol. 6421, pp. 320–329. Springer, Heidelberg (2010)
23. Choiński, D., Metzger, M., Nocoń, W.: Multiscale three-phase flow simulation dedicated to model based control. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part II. LNCS, vol. 5102, pp. 261–270. Springer, Heidelberg (2008)
24. Nocoń, W., Metzger, M.: Predictive Control of Decantation in Batch Sedimentation Process. *AICHE Journal* 56, 3279–3283 (2010)
25. Ni, B.-J., Fanf, F., Rittmann, B.E., Yu, H.-Q.: Modeling Microbial Products in Activated Sludge under Feast-Famine Conditions. *Environ. Sci. Technol.* 43, 2489–2497 (2009)
26. Cassandras, C.G., Lafortune, S.: Introduction to discrete event system. Springer, New York (2008)
27. Partalas, I., Feneris, I., Vlahavas, I.: Multi-Agent Reinforcement Learning using Strategies and Voting. In: 19th IEEE International Conference on Tools with Artificial Intelligence, pp. 318–324 (2007)
28. Gołacki, M., Koźlak, J., Żabińska, M.: Holonic-Based Environment for Solving Transportation Problems. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) *HoloMAS 2009*. LNCS, vol. 5696, pp. 193–202. Springer, Heidelberg (2009)
29. Metzger, M., Choinski, D., Nocon, W.: Pilot switchable SBR and SOCP processes for biological wastewater treatment. Activity Report – Institute of Automatic Control, Gliwice (2010)
30. National Instruments documentation, Using the LabVIEW Shared Variable, <http://zone.ni.com/devzone/cda/tut/p/id/4679>
31. Nocoń, W.: Requirement Specification for Agent-Based Cooperative Control of Dynamical Systems. In: Luo, Y. (ed.) *CDVE 2010*. LNCS, vol. 6240, pp. 270–277. Springer, Heidelberg (2010)

Active Sets and Unification

Germano Resconi¹ and Chris Hinde²

¹ Dept. of Mathematics and Physics,
Catholic University Brescia,
I-25121,
Italy
`resconi@numerica.it`

² Department of Computer Science,
Loughborough University,
Leicestershire LE11 3TU
UK
`c.j.hinde@lboro.ac.uk`

Abstract. An active set is a unifying space being able to act as a “bridge” for transferring information, ideas and results between distinct types of uncertainties and different types of applications. An active set is a set of agents who independently deliver *true* or *false* values for a given proposition. An active set is not a simple vector of logic values for different propositions, the results are a vector but the set is not. The difference between an ordinary set and active set is that the ordinary set has passive elements with values of the attributes defined by an external agent, in the active set any element is an agent that internally defines the value of a given attribute for a passive element. In this paper we show the connection between classical, fuzzy, evidence theory and active sets. In conclusion at one external agent we substitute a set of experts or agents that evaluate in a conflicting way the logic value of a given proposition or attribute. Under the same meta level of active sets we can discover analogy and similarity among distinct theories of uncertainty and types of many valued logics.

1 Concepts and Definitions

An active set is a set of agents which assign values in $\{true, false\}$ [7] to propositions. So for any set of agents we have the power set of all possible evaluations for the proposition p .

Formally we have:

$$\begin{aligned} SS(p) &= \{A, \Omega(p) \mid A = \text{set of agents,} \\ &\quad \Omega(p) = \text{power set } 2^{|\mathcal{A}|} \text{ of the evaluations}\} \end{aligned} \quad (1)$$

Given the proposition p , we could use criterion C to evaluate p in the set $\Omega(p)$. So with three agents we have eight criteria to evaluate the proposition:

$$\begin{aligned} \Omega(p, C_1) &= \begin{pmatrix} Agent & 1 & 2 & 3 \\ Logic & value & true & true & true \end{pmatrix} \\ \Omega(p, C_2) &= \begin{pmatrix} Agent & 1 & 2 & 3 \\ Logic & value & true & true & false \end{pmatrix} \\ \Omega(p, C_{\dots}) &= \begin{pmatrix} Agent & 1 & 2 & 3 \\ Logic & value & \dots & \dots & \dots \end{pmatrix} \end{aligned}$$

The set of criteria form a lattice.

1.1 Operations

The agents set A has the normal intersection union and complementary operators. For the logic evaluation we have two different operations.

1) Operations among criteria for the same proposition. Because we have the same proposition with two different criteria, we cannot compose logic values that are heterogeneous. So we have the rule

$$C_i \oplus C_j = \begin{pmatrix} Agent & 1 & 2 & \dots & n \\ C_i & v_{1,1} & v_{1,2} & \dots & v_{1,n} \\ C_j & v_{2,1} & v_{2,2} & \dots & v_{2,n} \end{pmatrix} \tag{2}$$

So we increase the dimension of the space of the evaluation. For example given six agents and two criteria we have

$$\Omega(p, C_i, C_j) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ C_i & f & t & t & t & f & f \\ C_j & f & f & t & t & t & t \end{pmatrix} \tag{3}$$

2) For two propositions p and q we have the composition rule:

$$\begin{aligned} \Omega(p \wedge q, C) &= \begin{pmatrix} Agent & 1 & 2 & \dots & n \\ p, C & v_{p,1} & v_{p,2} & \dots & v_{p,n} \end{pmatrix} \wedge \begin{pmatrix} Agent & 1 & 2 & \dots & n \\ q, C & v_{q,1} & v_{q,2} & \dots & v_{q,n} \end{pmatrix} \\ &= \begin{pmatrix} Agent & 1 & 2 & \dots & n \\ p, q, C & v_{p,1} \wedge v_{q,1} & v_{p,2} \wedge v_{q,2} & \dots & v_{p,n} \wedge v_{q,n} \end{pmatrix} \end{aligned} \tag{4}$$

Similarly for \vee .

Example

$$\begin{aligned} \Omega(p) &= \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Values & t & t & t & f & f & f \end{pmatrix}, & \Omega(q) &= \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Value & t & t & t & t & f & f \end{pmatrix}, \\ \Omega(p \wedge q) &= \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Value & t & t & t & t & f & f \end{pmatrix}, & \Omega(p \vee q) &= \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Value & t & t & t & t & f & f \end{pmatrix} \end{aligned} \tag{5}$$

The two logic operators are affected by the order of the agents
For the negation operator we have

$$\Omega(\neg p) = \begin{pmatrix} Agent & 1 & & & & & 2 \\ Value & \alpha_1(\neg v_1) + (1 - \alpha_1)(v_1) & \alpha_2(\neg v_2) + (1 - \alpha_2)(v_2) \\ & \dots & n \\ & \dots & \alpha_n(\neg v_n) + (1 - \alpha_n)(v_n) \end{pmatrix} \quad (6)$$

Example

$$\Omega(p) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Values & t & t & t & f & f & f \end{pmatrix} \quad (7)$$

$$if \alpha = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Value & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} then \Omega(\neg p) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Values & t & t & t & f & f & f \end{pmatrix} \quad (8)$$

$$if \alpha = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Value & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} then \Omega(\neg p) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 \\ Values & t & t & t & t & f & f \end{pmatrix} \quad (9)$$

When all the values of α are equal to one all the agents change their value in the negation operation. When one α is zero for the *true* value one *true* values agent does not change and all the others change. So at the end the number of agents with value equal to *true* in the negation operation is more than in the classical negation for any agent. At the reverse if α is zero for one agent with *false* value the number of the *true* value in the negation is less of the classical negation for any agent.

3) Aggregation rule for active set

Given an active set we associate to any active set evaluation a number by an aggregation function that can be linear or non linear. For the linear case the aggregation can be simple aggregation or can be weighted aggregation. For example for simple linear aggregation we have the rule.

$$\Omega(p, C_1) = \begin{pmatrix} Agent & 1 & 2 & 3 \\ Value & true & true & true \end{pmatrix} \quad (10)$$

$$\mu(p, C_1) = \frac{1}{3} | true + \frac{1}{3} | true + \frac{1}{3} | true = \frac{1}{3} + \frac{1}{3} + \frac{1}{3} = 1 \quad (11)$$

$$\Omega(p, C_2) = \begin{pmatrix} Agent & 1 & 2 & 3 \\ Value & true & true & false \end{pmatrix} \quad (12)$$

$$\mu(p, C_2) = \frac{1}{3} | true + \frac{1}{3} | true + \frac{1}{3} | false = \frac{1}{3} + \frac{1}{3} + \frac{0}{3} = \frac{2}{3} \quad (13)$$

2 Fuzzy Set by Active Set

The probability calculus does not explicitly incorporate the concepts of irrationality or agent’s state of logic conflict. It misses structural information at the level of individual objects, but preserves global information at the level of a set of objects. Given a die, probability theory studies frequencies of the different faces $E=\{e\}$ as independent events. This set of elementary events E has

no structure. It is only required that elements of E are *mutually exclusive* and *complete*, that is no other alternative is possible. The order of its elements is irrelevant to probabilities of each element of E . No irrationality or conflict is allowed in this definition relative to mutual exclusion. The classical probability calculus does not provide a mechanism for modelling uncertainty when agents communicate (collaborate or conflict). Below we present the important properties of sets of conflicting agents at one dimension [8]. Let $\Omega(x)$ the active set for the proposition x and $|\Omega(x)|$ be the numbers of agents for which proposition x is *true* then given two propositions p and q we order them in ascending order of cardinality.

$$|\Omega(p)| = 4, |\Omega(q)| = 5 \tag{14}$$

$$\Omega(p) = \left(\begin{array}{c} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ \text{Values } f\ f\ t\ t\ t\ t\ f\ f \end{array} \right), \Omega(q) = \left(\begin{array}{c} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ \text{Values } f\ t\ t\ f\ t\ t\ f\ t \end{array} \right)$$

$$|\Omega(p)| = 4, |\Omega(q)| = 5 \tag{15}$$

We have

$$\Omega(p \wedge q) = \left(\begin{array}{c} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ \text{Values } f\ f\ t\ f\ t\ t\ f\ f \end{array} \right), \Omega(p \vee q) = \left(\begin{array}{c} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ \text{Values } f\ t\ t\ t\ t\ t\ f\ t \end{array} \right)$$

$$|\Omega(p \wedge q)| = 3, |\Omega(p \vee q)| = 6 \tag{16}$$

Now we know that

$$q \vee (p \wedge \neg q) = (q \vee p) \wedge (q \vee \neg q) = q \vee p \tag{17}$$

$$p \wedge (p \wedge \neg q) = p \wedge (\neg p \vee q) = p \wedge \neg p \vee (p \wedge q) = p \wedge q \tag{18}$$

But because when q is *false* and p is *true* we adjoin at q one logic value *true* to obtain p or q . So when we repeat many times for any agent this process we have that at the number of *true* values for q we must adjoin other *true* values for which q is *false* but p is true. In conclusion we have

$$|\Omega(p \vee q)| = |\neg q \wedge p| = \max(|\Omega(q)|, |\Omega(q)| + |\Omega(\neg q \wedge p)|) \tag{19}$$

For and operation we have that when q is *false* and p is *true* we eliminate one element for which p is true. In conclusion when we repeat this for many times we have

$$|\Omega(p \wedge q)| = |\Omega(q)| - |\neg q \wedge p| = \min(|\Omega(q)|, |\Omega(q)| - |\Omega(\neg q \wedge p)|) \tag{20}$$

Zadeh rule

$$|\Omega(p \wedge q)| = \min(|\Omega(p)|, |\Omega(q)|, |\Omega(p \vee q)|) = \max(|\Omega(p)|, |\Omega(q)|) \tag{21}$$

So when the agents for which p is *true* are also the agents for which q is true. We can see that for the previous example we have

$$\begin{aligned}
 |\Omega(p \wedge \neg q)| &= 1 \\
 |\Omega(p \wedge q)| &= \min(|\Omega(p)|, |\Omega(q)|) - |\Omega(p \wedge \neg q)| = 4 - 1 = 3 \\
 |\Omega(p \vee q)| &= \max(|\Omega(p)|, |\Omega(q)|) + |\Omega(p \wedge \neg q)| = 5 + 1 = 6 \quad (22)
 \end{aligned}$$

For the negation we have the Zadeh rule

$$|\Omega(\neg p)| = n - |\Omega(p)| \quad (23)$$

When we divide by n, the number of agents, we have the traditional rule

$$\mu(\neg p) = \frac{|\Omega(\neg p)|}{n} = 1 - \mu(p) \quad (24)$$

In this situation all the agents in the negation change in a synchronic way all the logic value. But when we have the Sugeno rule

$$|\Omega(\neg p)| = \mu(\neg p)n = \frac{1 - \mu(p)}{1 + \lambda\mu(p)}n = \frac{1 - \frac{|\Omega(p)|}{n}}{1 - \lambda\frac{|\Omega(p)|}{n}}n = n\frac{n - |\Omega(p)|}{n + \lambda|\Omega(p)|} \quad (25)$$

When $\lambda = 0$ all the agents change their logic value synchronously. So before we have three *true* values and three *false* values for the negation we have again the same values but reversed. For

$$\begin{aligned}
 \text{if } \lambda = 0, |\Omega(\neg p)| &= n - |\Omega(p)| = 6 - 3 = 3 \\
 \text{if } \lambda < 0, |\Omega(\neg p)| &> n - |\Omega(p)| \\
 \text{if } \lambda > 0, |\Omega(\neg p)| &< n - |\Omega(p)| \quad (26)
 \end{aligned}$$

When λ is negative agents with *true* values do not change, when λ is positive agents with *false* logic values do not change. In conclusion t-norm and t-conorm and fuzzy negation can be emulated inside the active set.

3 Evidential Reasoning

In the evidence theory any subset $A \subseteq \mathbb{U}$ of the universe \mathbb{U} is associated with a value $m(A)$ called a basic assignment probability such that:

$$\sum_{k=1}^{2^N} m(A_k) = 1$$

Respectively, the belief $Bel(\Omega)$ and plausibility $Pl(\Omega)$ measures for any set Ω are defined as:

$$Bel(\Omega) = \sum_{A_k \subseteq \Omega} m(A_k), Pl(\Omega) = \sum_{A_k \cap \Omega \neq \emptyset} m(A_k) \quad (27)$$

Baldwin’s theory of fuzzy sets [1] relates the domains of probability, possibility and fuzzy sets. Mass assignments are related directly to fuzzy sets through Baldwin’s voting interpretation of fuzzy sets [2]. Using the voting interpretation

any normalized fuzzy set can be expressed as a mass assignment [3]. Equation 28 shows a simple discrete fuzzy set.

$$F = \{1 \mid a, 0.7 \mid b, 0.5 \mid c, 0.1 \mid d\} \tag{28}$$

The mass assignment m is derived from F using the voting interpretation. In this example we consider a group of 10 voters, each of whom must vote on the acceptance of each member of X given fuzzy set F . Normalizing the number of voters accepting any one proposition generates the mass assignment m for fuzzy set F shown in Equation 29.

$$m = \{a\} : 0.3, \{a, b\} : 0.2, \{a, b, c\} : 0.4, \{a, b, c, d\} : 0.1 \tag{29}$$

We can now use the calculus of mass assignment to reason with fuzzy sets at the mass level. The advantage of this representation is the close relationship between mass assignments and their corresponding families. Using aspects of inconsistency it is possible to use active sets and the ensuing inconsistencies to derive new knowledge [5].

3.1 Mass Assignment and Active Set

In ACTIVE SET theory at any possible vote we associate an agent that can evaluate the vote to True or False. For the fuzzy distribution we have the four possible epistemic logic evaluations for each agent.

$$K_\alpha(a \in A) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ Vote & t & t & t & t & t & t & t & t & t & t \end{pmatrix} \tag{30}$$

For all the 10 agents $a \in A$. Now we repeat for all the other elements of the fuzzy set A . So we obtain:

$$K_\alpha(c \in A) = \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ Vote & t & t & t & t & t & f & f & f & f & f \end{pmatrix} \tag{31}$$

For 5 agents $c \in A$ and for 5 agents $b \notin A$. So the agents disagree one with another

Now we can compare the ACTIVE SET with the absolute True set and the absolute False set. In fact we have

$$\begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ Vote & t & t & t & t & t & t & t & t & t & t \end{pmatrix} = \{T\}, \begin{pmatrix} Agent & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ Vote & f & f & f & f & f & f & f & f & f & f \end{pmatrix} = \{F\} \tag{32}$$

Given the proposition

$p(a)$ = “the element a belongs to the set A ”

the negation

$\neg p(a)$ = “the element a does not belong to the set A ”

the sets $A = \{a\}$ and $\mathbb{U} = \{a, b, c, d\}$

for the set A we have the logic expression

$$q(A) = p(a) \wedge \neg p(b) \wedge \neg p(c) \wedge \neg p(d)$$

in fact in the set A we have only one element “ a ”, the other elements do not belong to the set A . Now in ACTIVE SET we have

$$p(a) = K_\alpha(a \in A)$$

The valuation of the sentence $p(a)$ is made by 10 agents. Any agent produces a vote which value is *true* ($\in A$) or *false* ($\neg \in A$).

$$\begin{aligned} m : \{a\} &= K_\alpha(a \in A) \wedge K_\alpha(b \notin A) \wedge K_\alpha(c \notin A) \wedge K_\alpha(d \notin A) \\ &= \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ t \ t \ t \ t \ t \ t \ t \ t \ t \end{array} \right) \wedge \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ f \ f \ t \ t \ t \end{array} \right) \wedge \\ &\quad \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ t \ t \ t \ t \ t \end{array} \right) \wedge \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ t \ t \ t \ t \ t \ t \ t \ t \ t \end{array} \right) \\ &= \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ f \ f \ t \ t \ t \end{array} \right) \\ &= \frac{3}{10} = 0.3 \end{aligned} \quad (33)$$

Similarly for other elements in the power set; in conclusion we have

$$m = \{a\} : 0.3, \{a, b\} : 0.2, \{a, b, c\} : 0.4, \{a, b, c, d\} : 0.1 \quad (34)$$

The masses for all the other set in the power set are equal to zero. The Belief and Plausibility measures can be obtained by active set.

$$\begin{aligned} \text{Bel} : \{a, b\} &= m_A \{a\} + m_A \{b\} + m_A \{a, b\} \\ &= K_\alpha(a \in A) \wedge K_\alpha(b \notin A) \wedge K_\alpha(c \notin A) \wedge K_\alpha(d \notin A) \vee \\ &\quad K_\alpha(a \notin A) \wedge K_\alpha(b \in A) \wedge K_\alpha(c \notin A) \wedge K_\alpha(d \notin A) \vee \\ &\quad K_\alpha(a \in A) \wedge K_\alpha(b \in A) \wedge K_\alpha(c \notin A) \wedge K_\alpha(d \notin A) \\ &= \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ f \ f \ t \ t \ t \end{array} \right) \vee \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ f \ f \ f \ f \ f \end{array} \right) \vee \\ &\quad \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ t \ t \ f \ f \ f \end{array} \right) \\ &= \left(\begin{array}{c} \text{Agent } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{Vote } \ f \ f \ f \ f \ f \ t \ t \ t \ t \ t \end{array} \right) \\ &\Rightarrow \frac{5}{10} = 0.3 + 0.0 + 0.2 = 0.5 \end{aligned} \quad (35)$$

Similarly for the plausibility measure

3.2 Semantic unification and separation with logic implication

Hinde [64] and others define semantic unification (SU) as a method of comparing how well one mass assignment supports another, as with the conditional

probability. In the unification process we start with Y as the universal set and X as a conditional set. Now with the definition of the elements $p(a)$, $p(b)$, $p(c)$

$$\begin{aligned} p(a_X) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ t\ t\ t\ f \end{pmatrix} \\ p(b_X) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \\ p(c_X) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ f\ f\ f\ f\ f \end{pmatrix} \end{aligned} \quad (36)$$

and

$$\begin{aligned} p(a_Y) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ t\ t\ t\ f \end{pmatrix} \\ p(b_Y) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \\ p(c_Y) &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ f\ f\ f\ f\ f \end{pmatrix} \end{aligned} \quad (37)$$

In classical logic the equivalent to implication is

$$p(A) \rightarrow p(B) \equiv \neg p(A) \vee p(B)$$

so we have

$$\begin{aligned} p(a_X) \rightarrow p(a_Y) &\equiv \text{True} \\ &\equiv \neg p(a_X) \vee p(a_Y) \\ &\equiv \neg \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ t\ t\ t\ f \end{pmatrix} \vee \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \\ &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } f\ f\ f\ f\ f\ f\ f\ f\ f\ t \end{pmatrix} \vee \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ t\ t\ t\ f \end{pmatrix} \\ &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ t\ t\ t\ t \end{pmatrix} \\ \text{For "False" } \Rightarrow \mu(\text{True}) &= \frac{10}{10} = 1.0 \end{aligned} \quad (38)$$

$$\begin{aligned} p(b_X) \rightarrow \neg p(b_Y) &\equiv \text{False} \equiv \neg p(b_X) \vee \neg p(b_Y) \\ &\equiv \neg \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } f\ f\ f\ f\ f\ f\ t\ t\ t\ t \end{pmatrix} \vee \neg \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } f\ f\ f\ f\ f\ f\ t\ t\ t\ t \end{pmatrix} \\ &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \vee \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \\ &= \begin{pmatrix} \text{Agent } 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10 \\ \text{Vote } t\ t\ t\ t\ t\ t\ f\ f\ f\ f \end{pmatrix} \Rightarrow \mu(\text{False}) = \frac{6}{10} = 0.6 \end{aligned} \quad (39)$$

For the two values (False, True) we have

$$\begin{aligned}
p(a_X) \rightarrow p(a_Y) &\equiv \text{True} \quad , \quad p(b_X) \rightarrow \neg p(b_Y) \equiv \text{False} \\
&\text{or} \\
\begin{pmatrix} \text{True} & p(a_X) \rightarrow p(a_Y) \\ \text{False} & p(b_X) \rightarrow \neg p(b_Y) \end{pmatrix} &= \begin{pmatrix} \text{Agent} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \text{True} & t & t & t & t & t & t & t & t & t & t \\ \text{False} & t & t & t & t & t & t & f & f & f & f \end{pmatrix} \\
\mu(\text{True}, \text{False}) &= \frac{6}{10} = 0.6 \tag{40}
\end{aligned}$$

For 6 agents the valuation True and False are true.

We can have also the vector evaluation (True, False, False) for which we have

$$\begin{aligned}
p(a_X) \rightarrow p(a_Y) &\equiv \text{True} \\
p(b_X) \rightarrow \neg p(b_Y) &\equiv \text{False} \\
p(c_X) \rightarrow \neg p(c_Y) &\equiv \text{False} \\
&\text{or} \\
\begin{pmatrix} \text{True} & p(a_X) \rightarrow p(a_Y) \\ \text{False} & p(b_X) \rightarrow \neg p(b_Y) \\ \text{False} & p(c_X) \rightarrow \neg p(c_Y) \end{pmatrix} , \quad \begin{pmatrix} \text{Agent} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \text{True} & t & t & t & t & t & t & t & t & t & t \\ \text{False} & t & t & t & t & t & t & f & f & f & f \\ \text{False} & t & t & t & t & t & f & f & f & f & f \end{pmatrix} \\
\mu(\text{True}, \text{False}) &= \frac{5}{10} = 0.5 \tag{41}
\end{aligned}$$

4 Empty Set Structure and Active Set

Given the simple empty set $\{\}_\alpha$ it can be evaluated by the active set in this way

$$\begin{aligned}
p(\{\}_\alpha) &= K_\alpha(a_1 \notin A) \wedge K_\alpha(a_2 \notin A) \wedge \cdots \wedge K_\alpha(a_n \notin A) \\
A &= \{a_1, a_2, \dots, a_n\} \tag{42}
\end{aligned}$$

Where α is the criteria that the agent uses to evaluate the proposition.

For the complex empty set $\{\{\}_\alpha\}_\beta$ we use the knowledge operator $K_\beta K_\alpha$ so we have

$$p(\{\{\}_\alpha\}_\beta) = K_\beta[K_\alpha(a_1 \notin A) \wedge K_\alpha(a_2 \notin A) \wedge \cdots \wedge K_\alpha(a_n \notin A)] \tag{43}$$

For the other type of empty set as

$\{\{\}_\alpha, \{\}_\beta\}$

We have

$$p(\{\{\}_\alpha, \{\}_\beta\}) = p(\{\}_\alpha) \wedge p(\{\}_\beta)$$

With the complex K epistemic operator in active set we can evaluate any type of complex empty set.

5 Conclusion

We present properties and operations for active set. The aims are: first to show that uncertainty in different forms are models of active sets; second to avoid linguistic approach to uncertainty and to focus attention on the agent as source of uncertainty; third is a simplification which aims to establish a bridge between local classical evaluation to the many valued logic of fuzzy logic. In the active set we do not have a total separation between classical logic and other types of logics. An agent does not have complete knowledge of the proposition but only local knowledge so it cannot evaluate a proposition universally. Other agents can evaluate the same proposition differently. The lack of knowledge creates conflicts, uncertainties, inconsistency *true* contradiction, *false* tautology that are represented by a complex evaluation structure.

References

1. Baldwin, J.: A theory of mass assignments for artificial intelligence. In: Driankov, D., L. Ralescu, A., Eklund, P. (eds.) IJCAI-WS 1991. LNCS, vol. 833, pp. 22–34. Springer, Heidelberg (1994)
2. Baldwin, J.: A calculus for mass assignments in evidential reasoning. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 513–531. John Wiley & Sons, Inc., New York (1994)
3. Baldwin, J., Martin, T., Pilsworth, B.: Fril - Fuzzy and Evidential Reasoning in Artificial Intelligence. Research Studies Press Ltd., Taunton (1995)
4. Hinde, C., Patching, R., Stone, R., Xhemali, D., McCoy, S.: Reasoning Consistently about Inconsistency. In: Garibaldi, J., Angelov, P. (eds.) Proceedings of 2007 IEEE International Conference on Fuzzy Systems, pp. 769–775 (2007)
5. Hinde, C., Resconi, G.: Active sets and knowledge acquisition. In: 5th International KES Conference on Agents and Multi-agent Systems Technologies and Applications 2011. IEEE, Manchester (2011)
6. Patching, R., Hinde, C., McCoy, S.: Inconsistency and semantic unification. Fuzzy Sets and Systems 157, 2513–2539 (2006)
7. Resconi, G., Hinde, C.: Active sets, fuzzy sets and inconsistency. In: Aranda, J., Xambo, S. (eds.) Proceedings of FUZZIEEE 2010, pp. 354–357. IEEE, Los Alamitos (2010)
8. Resconi, G., Kovalerchuk, B.: Agents in Quantum and Neural Uncertainty. Medical Information Science Reference (Hershey, New York) (2011)

Evaluating Interaction of MAS Providing Context-Aware Services

Nayat Sanchez-Pi, David Griol, Javier Carbo, and Jose M. Molina

Carlos III University of Madrid,
Avda de la Universidad Carlos III, 22. Colmenarejo. Madrid
{nayat.sanchez,david.griol,javier.carbo,
josemanuel.molina}@uc3m.es

Abstract. Much effort has been spent on suggesting and implementing new architectures of MAS to specific domains. Often each new architecture is not even compared to any existing architectures in order to evaluate their potential benefits. The evaluation of Multi-Agent Systems (MAS) is a complex problem and it does not have a single form. The present work follows the research line of considering the agent interaction as the main evaluation criteria, the most important characteristic of any complex software as autonomous agents according to [9]. So, in this paper, we have suggested an assignment of evaluation values to Agents interaction in a specific MAS architecture for providing context services by means of conversational agents. This evaluation is mainly based on the relevance of the messages content brought by an interaction. For dependant nature of the relevance of the messages, the valuation has to be adhoc, but our paper provides an example of how interesting is this alternative in order to evaluate any MAS architecture theoretically.

Keywords: multi-agent systems, evaluation.

1 Introduction

The evaluation of MAS makes possible two main objectives. Firstly, to understand their behavior and secondly, to compare the operation of several systems. In the literature there are works addressing this evaluation based on the architectural style [1]; software engineering related criteria and characteristics of MAS [7]; [4], or the complexity of interactions [6]. The evaluation of the distributed nature of MAS and the complexity of the interaction inside them is a very difficult task. The consideration of the interaction as the most important characteristic of MAS [9], allows studying and comparing this kind of systems at the level of their interactions. In this paper, we suggest an assignment of evaluation values to Agents interaction in a specific MAS architecture for providing context services. This evaluation is mainly based on the relevance of the messages content brought by an interaction. For dependant nature of the relevance of the messages, the valuation has to be adhoc, but our paper provides an example of how interesting is this alternative in order to evaluate any MAS architecture theoretically. The rest of the paper is structured as follows. Section 2 briefly describes our agent-based architecture to provide context-aware services. Section 3 details the evaluation method based on the

weight of the information specified for each message. Section 4 presents the application of the proposed evaluation methodology to a specific scenario. Finally, we draw some conclusions and suggest some future directions of research.

2 Our Agent System to Provide Context-Aware Services

The proposed agent-based architecture manages context information to provide personalized services by means of users interactions with conversational agents. As it can be observed in Figure 1, it consists of five different types of agents that cooperate to provide an adapted service. *User agents* are configured into mobile devices or PDAs. *Provider Agents* are implemented by means of *Conversational Agents* that provide the specific services. A *Facilitator Agent* links the different positions to the providers and services defined in the system. A *Positioning Agent* communicates with the ARUBA positioning system [8] to extract and transmit positioning information to other agents in the system. Finally, a *Log Analyzer Agent* generates user profiles that are used by Conversational Agents to adapt their behaviour taking into account the preferences detected in the users' previous dialogues.

A conversational agent is a software that accepts natural language as input and generates natural language as output, engaging in a conversation with the user. To successfully manage the interaction with the users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialogue management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS). These tasks are usually implemented in different modules. In our architecture, we incorporate a Context Manager in the architecture of the designed conversational agents, This module deals with loading the context information provided

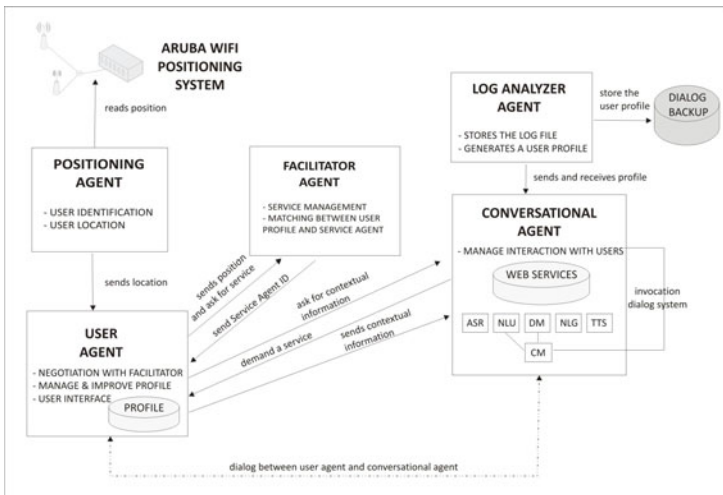


Fig. 1. Schema of the proposed multi-agent architecture

by the User and Positioning Agents, and communicates it to the different modules of the Conversational Agent during the interaction.

To manage context information we have defined a data structure called *user profile*. Context information in our user profile can be classified into three different groups. *General user information* stores user's name and machine identifier, gender, preferred language, pathologies or speech disorders, age, *Users Skill level* is estimated by taking into account variables like the number of previous sessions, dialogues and dialogue turns, their durations, time that was necessary to access a specific web service, the date of the last interaction with the system, etc. Using these measures a low, medium, normal, high or expert level is assigned. *Usage statistics and preferences* are automatically evaluated taking into account the set of services most required by the user during the previous dialogues, date and hour of the previous interactions and preferred output modality.

The interaction with the different agents follows a process which consists of the following phases:

1. The ARUBA positioning system is used to extract information about the positions of the different agents in the system. This way, it is possible to know the positions of the different User Agents and thus extract information about the Conversational Agents that are available in the current location.
2. The Positioning Agent reads the information about position (coordinates x and y) and place (*Building* and *Floor*) provided by the ARUBA Positioning Agent by reading it from a file, or by processing manually introduced data.
3. The Positioning Agent communicates the position and place information to the User Agent.
4. Once a User Agent is aware of its own location, it communicates this information to the Facilitator Agent in order to find out the different services available in that location.
5. The Facilitator Agent informs the User Agent about the services available in this position .
6. The User Agent decides the services in which it is interested.
7. Once the User Agent has selected a specific service, it communicates its decision to the Facilitator Agent and queries it about the service providers that are available.
8. The Facilitator Agent informs the User Agent about the identifier of the Conversational Agent that supplies the required service in the current location.
9. The User Agent asks the Conversational Agent for the required service.
10. Given that the different services are provided by context-aware Conversational Agents, they ask the User Agent about the context information that would be useful for the dialogue. The User Agent is never forced to transmit its personal information and preferences. This is only a suggestion to customize the service provided by means of the Conversational Agent.
11. The User Agent provides the context information that has been required.
12. The conversational agent manages the dialogue providing an adapted service by means of the context information that it has received.
13. Once the interaction with the Conversational Agent has finished, the Conversational Agent reads the contents of the log file for the dialogue and send this information to the Log Analyzer Agent.

14. The Log Analyzer Agent stores this log file and generates a user profile to personalize future services. This profile is sent to the Conversational Agent.

In our architecture we need an objective description of the concepts and relationships of the domain of knowledge of the messages exchange by agents. This explicit and formal specification of a shared conceptualization is what we usually called ontology [5]. An ontology allows that a content of a message can be interpreted unambiguously and independently from the context.

Eight concepts have been defined for the ontology of the system. The definition is: *Location* (*XCoordinate* int, *YCoordinate* int), *Place* (*Building* int, *Floor* int), *Service* (*Name* String), *Product* (*Name* String, *Characteristics* List of Features), *Feature* (*Name* String, *Value* String), *Context* (*Name* String, *Characteristics* List of Features), *Profile* (*Name* String, *Characteristics* List of Features), *DialogLog* (*Log* List of Strings).

Our ontology also includes six predicates with the following arguments: *HasLocation* (*Place*, *Position*, and *AgentID*), *HasServices* (*Place*, *Position*, and *List of Services*), *isProvider* (*Place*, *Position*, *AgentID*, *Service*), *HasContext* (*What*, *Who*), *HasDialog* (*DialogLog* and *AgentID*), *HasProfile* (*Profile* and *AgentID*), and *Provide* (*Product* and *AgentID*).

The free software JADE (Java Agent Development Framework)¹ has been used for the implementation of our architecture. It was the most convenient option as it simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines and the configuration can be controlled via a remote GUI.

3 Evaluation Proposal Based on Agents Interaction

The consideration of interaction as the main of evaluation has been addressed by other researches. These studies have verified that this kind of evaluation originates different types of problems [6]. Firstly, the effect of an interaction unit (a single message) in an agent system could be equivalent to the definition of n units (messages) in another system. This way, the weight assigned to the same interaction is 1 in the first system and it n in the second. Secondly, the interaction units that are received and cannot be used by an agent could be a bias in the measurement of interaction in MAS. Our proposal is based on [6]. The first task is to classify the possible received messages into specific sets sharing the same type. Then, a weight is associated to a message according to its type. If two messages with the same type produce very different effects on the agent, then this assignment does not provide a correct solution. The effects of considering interactions as the main feature of the evaluation, consists of initially processing a message and then decide a responsive action. The initial processing is carried out in two phases, which consist of the memorization that deals with the change at the internal state caused by the received message, and the decision that concern the choice of the action that will be handled. According to the evaluation model, two kinds of functions are considered:

¹ <http://jade.tilab.com/>

- A function *Interaction* associates a weight to each message according to its type. This function can be computed adopting the primitives proposed by [3] to the type of interaction. This work describes four possibilities kinds of messages: present, request, answer, and inform. These four types have to be distinguished due to the different basic behaviors that they model from the sender or the receiver points of view. Therefore, if M_{sent}^A : the set of messages sent by agent A and if $M_{received}^A$: the set of messages which may be received by agent A, the function *Interaction* associates for a message sent by the agent A, a message received by the agent B:

$$Interaction = M_{sent}^A \rightarrow M_{received}^B \quad (1)$$

- This solution partially resolves the problem. However, it is only valid if two messages of the same type have equivalent effects on the agent. This way, we introduce the function Φ [6] to associate a different weight to a message according to the change on the internal state and the actions triggered after its reception. This function evaluates the different effect of a message in agent systems. For better understanding Φ is divided into two terms: the first one evaluates Decision, DD^A and the second one evaluates Memorization, MM^A . The term MM^A associates a value to the variation of the internal state (caused by memorization step). To quantify these two terms, some measurable characteristics of the internal state must be defined. The specification of these characteristics is related to each specific application domain. The variation on one of these characteristics implies the function MM^A is considered as the sum of these weights:

$$MM^A = M_{received}^A \times S^A \rightarrow S^A \quad (2)$$

With regard to DD^A , this term associates a value to the triggered actions (i.e., results of decision step). To quantify this term, different types of actions must be defined and associate a weight to each of them. Then, the value of the function DD^A is calculated as the sum of the weights assigned to the triggered actions. Let S^A be the set of possible internal states for the agent A and let A^A be the set of actions may be done by agent A, then:

$$DD^A = M_{received}^A \times S^A \rightarrow A^A \quad (3)$$

Finally, the function Φ is defined as the sum of these functions DD^A and MM^A and the evaluation of the interactions in the MAS is based on the combination of the two functions *Interaction* + Φ .

4 Evaluation Our the Proposed MAS Architecture

In this section we present the application of the evaluation method described in the previous section o our context-aware agent architecture. We compute the described evaluation functions and assign different weights to each message in the agents interaction. In section 4.1 we define the *Interaction* function and in section 4.2, the Φ function is described.

4.1 Weights vs Type of Message: Function Interaction

The Interaction function according to the four message types described in the previous section (present, request, answer, and inform). Following, we detail the different phases during the interaction of the different agents and the different messages that are generated for the provision of the service. Three types of messages have to be distinguished because of the different basic behaviors that they model from the sender or the receiver points of view:

- A request includes a change of state of the sender, waiting for the answer.
- An inform includes no change of state for both the sender and the receiver. It might generate other informs, and possibly answers.
- A present includes a possible change in the state of the sender and/or of the receiver. Typically, a present will enable entering a society and introduce itself to other agents

Services are offered by means of a connection with the conversational agent.

1. The ARUBA positioning system notices a change in the position of a given User agent: *no message involved*
2. The Positioning Agent reads the information about position provided by the ARUBA Positioning Agent in the corresponding file: *no message involved*
3. The Positioning Agent communicates the information about position (coordinates x and y) and place (Building and Floor) to the User Agent: *present message*.
4. Once a User Agent knows its location, it asks the Facilitator Agent about the different services that are available in that location : request message.
5. The Facilitator Agent informs the User Agent about the available services: *inform message*
6. The User Agent then decides the services in which it is interested: *no message involved*
7. Once the User Agent has selected a specific service, it communicates its decision to the Facilitator Agent and ask it about the service providers that are available: *request message*
8. The Facilitator Agent informs the User Agent about the identifier of the Conversational Agent that supplies the required service in the current location: *inform message*
9. The User Agent asks the Conversational Agent for the required service: *request message*
10. The Conversational Agent asks the User Agent about the context information that would be used to adapt the provided service: request message
11. The User Agent provides the context information that has been required: *inform message*
12. The conversational agent manages the dialogue providing an adapted service by means of the context information that has been received: *no message involved*
13. Once the interaction with the Conversational Agent has finished, this agent sends a log file to the Log Analyzer Agent: *inform message*

14. The Log Analyzer Agent stores this log file and generates a new user profile to personalize future services. This profile is sent to the Conversational Agent: *inform message*

Therefore we associate the next values to the message types:

- request: 2 (a change of state, a reaction produced)
- inform: 1 (no change of state)
- present: 1.5 (1 or 2 change of state)

In order to assign weights to each type of messages, we follow this criteria: if there is any interest over a negotiation from the Users side, then the maximum weight is assigned to the Agree-proposal or Reject-proposal messages; if it is the system who tries to recommend a service based on its behavior, the maximum weight is assigned to Inform-ref or Propose messages.

4.2 Weights vs Treatment of a Message: Function Φ

As stated in Section 3, the Φ function computes the variation of the internal state of the agent caused by a memorization step and also a decision step. Memorization is evaluated by means of the MM^A . The defined ontology allows to measure that the internal state has changed due to the number of concepts involved or the number of attributes involved. We could have also considered the different relevance of the set of attributes and concepts by assigning different weights to each of them. For instance:

- Phase 10: Changes in concepts and their attributes may have a low value. In this phase, the User Agent provides the context information. Since the User Agent is never forced to transmit its personal information and preferences, the relevance of this package is low. The weights that are assigned to the context information depend on the utility of concepts and attributes to achieve the provision of the required service during the dialogue interaction.
- Phase 3 and 12: The weights assigned to the changes produced by the messages in the phase 3 depend on the number of services that whether available (positive) or not (negative) in the specific location (position and place). In this phase, the Positioning Agent communicates the position and place information to the User Agent. The weights are assigned similarly to the changes produced by each message in the phase 12. In this phase, the Facilitator Agent informs the User Agent about the services that are available in the current position.
- Phase 13: The utility of the messages corresponding to this phase, in which the Conversational Agent sends the log file of the dialogue to the Log Analyzer Agent, depends on the coincidence of this log with regard to the previous dialogues (partly time-decrescent function).
- Phase 14: Finally, a high weight is assigned to the message of this phase, in which the Log Analyzer Agent sends a user profile to the Conversational Agent. Since this profile is generated using several dialogue logs to personalize the provided service, a high relevance is assigned to this message.

The term DD^A indicates the variation of the internal state due to a decision step. This function associates a value to each triggered action. Different types of actions must be defined to quantify this term, each having a specific weight. Then, the value of DD^A is calculated as the sum of the weights of the triggered actions. The set of actions involved in our agent architecture can be classified as external or internal. External actions involve communicative responses for the given message. The weight of this reactive action is equivalent to the weight of the content included in the received message. Internal actions involve the processing and decision making described in the following phases:

- Phase 4: Once a User Agent knows its location, it asks the Facilitator Agent about the different services available in that: request message. Simple query to the internal database of the Facilitator Agent. No intelligence involved: minimal weight.
- Phase 6: The User Agent decides the services in which it is interested. Intelligent and relevant decision with a real economic cost: maximal weight.
- Phase 8: The Facilitator Agent informs the User Agent about the identifier of the Conversational Agent that supplies the required service in the current location: Simple query to the internal database of Facilitator agent. No intelligence involved: minimal weight.
- Phase 11: The User Agent provides the context information that has been required. Intelligent and relevant decision with a privacy cost: maximal weight
- Phase 13: Once the interaction with the Conversational Agent has finished, the Conversational Agent sends the log file generated after the dialogue to the Log Analyzer Agent. Simple query to the internal database of Conversational agent. No intelligence involved: minimal weight.
- Phase 14: The Log Analyzer Agent stores the log file and updates the user profile to personalize future services. Intelligent and relevant decision: medium weight.

Then we need to compute DD^A function that associate the variation of internal state caused by decision step. This function associates a value to the triggered actions. To quantify, certain type of actions must be defined. A type of actions having a weight. Then, the value of the function DD^A is considered as the sum of the weights of triggered actions. The set of actions involved in our agent system can be classified as external and internal. Where the external actions means communicative responses to the given message, where the weight of this reactive action is equivalent to the weight of the content included in the responsive message. On the other hand, internal actions involve the processing and decision making of the next phases:

4.3 Practical Application: The Case of an Airport Domain

We have used a previously defined domain of an Airport [8] as a testbed of our proposal. To introduce an experimentation case we use two different architectures. We compare them using different types of messages for a common domain: Airport presented in [8]. In previous work we design an architecture that has three types of agents: Central Agent, Provider Agents and User Agents [2]. From now on we call it MAS-CENTRAL-AGENT. Later we adapt this architecture to cope with new functionalities including the speech based interface and that is the new architecture presented in Figure 1, from now on we call it: MAS-CONVERSATIONAL-AGENT. New agents functionalities in this

new architecture state as follow: *Positioning agent* main tasks rely on the user identification and user location into the environment. *Facilitator agent* is the responsible of the services management and the discovering of services agent identification. *Conversational Agents* provide the specific services. Finally, a *Log Analyzer Agent* generates user profiles that are used by Conversational Agents to adapt their behaviour taking into account the preferences detected in the users' previous dialogues.

If we compute the amount of messages exchanged of each type for each architecture, we could then draw some conclusions.

We take experimentation using "Service Recommendation" in an Airport Domain [8] for MAS-CENTRAL-AGENT and MAS-CONVERSATIONAL-AGENT. The comparison involves the following 5 type of messages: *Agree-proposal*; *Inform-ref*; *Propose*; *Query-if*; *Reject-proposal*; *Request*. For instance, once the passenger John Mayer is inside the Airport, the system recommends him with a SPA service based on the reputation. An example of message in FIPA is:

```
(inform-ref
:sender (agent-identifier :name SPA)
:receiver (set (agent-identifier :name john mayer))
:content
((action (agent-identifier :name SPA)
(try (spa $product $reputation))
:protocol fipa-request
:language FIPA-SL
:ontology airport-ontology
:reply-with try-spa)
```

In order to assign weights to each type of messages, we follow this criteria: if there is any interest over a negotiation from the Users side, then the maximum weight is assigned to the *Agree-proposal* or *Reject-proposal* messages; if it is the system who tries to recommend a service based on its behavior, the maximum weight is assigned to *Inform-ref* or *Propose* messages. In the case of MAS-CONVERSATIONAL-AGENT, the *Inform-ref* message belongs to Phase 12 of our MAS interaction and would have a maximal weight because it is the systems intention to recommend a service based on the behaviour of the passenger. In this case is that the passenger is tired. Then, following the evaluation method described above and used in [6], MAS-CONVERSATIONAL-AGENT in contrast to MAS-CENTRAL-AGENT. Among the MAS-CENTRAL-AGENT properties the reactivity, communication, robustness and scalability highlight apart from the others. On the other hand, the MAS1 is more robust when evaluating fairness and load balancing. In this sense, the difference between the two systems, when performing the comparison on the number of messages sent is minimal. Therefore, MAS-CONVERSATIONAL-AGENT is valid.

5 Conclusions

In this paper, we have detailed a generic interaction-based proposal for the evaluation of MAS. The evaluation of the interactions in Agent Systems makes possible to compare the various alternative agent architectures and protocols between them. The idea of considering agent interactions as evaluation benchmark is not new since it was proposed by [6] in a general form. In this paper, we have applied it and specified it in order

to evaluate our Multi-Agent System (MAS) that provides Context-Aware Services. We have assigned evaluation values according to the general idea of [6]. Additionally we gave weights to the exchanged messages by the agents in our MAS architecture. Finally, the example of the Airport domain let us experience the complexity of the evaluation problem and state clear criterion for weight assignments. As future work we would aim to include a further evaluation of our system using different execution instances in several domains with this technique. Although the original evaluation method (from [6]) is itself general, it needs adhoc adaptation to be applied in whatever agent system, so presenting a particular adaptation of this general method is interesting due to it facilitates the general understanding of the evaluation process of MAS, and it contributes to the general necessity of a theoretical comparison of different agent architecture approaches to any distributed problem.

Acknowledgments

Funded by projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM MADRINET S-0505/TIC/0255, and DPS2008-07029-C02-02.

References

1. Davidsson, P., Johansson, S., Svahnberg, M.: Characterization and evaluation of multi-agent system architectural styles. In: *Software Engineering for Multi-Agent Systems IV*, pp. 179–188 (2006)
2. V., Sanchez-Pi, N., Carbo, J., Molina, J.M.: Reputation in user profiling for a contextaware-multiagent system. In: *EUMAS (2006)*
3. Gaspar, G.: Communication and belief changes in a society of agents: Towards a formal model of autonomous agent, d.a.i. 2. In: Demazeau, Y., M.J.P.: (ed.) *Decentralized A. I. 2*, pp. 245–255. Elsevier Science, Amsterdam (1991)
4. Giunchiglia, F., Mylopoulos, J., Perini, A.: The tropos software development methodology: Processes, models and diagrams. In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 63–74. ACM Press, New York (2002)
5. Gruber, T.: The role of common ontology in achieving sharable, reusable knowledge bases. In: *2nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, USA, pp. 601–602 (April 1991)
6. Joumaa, H., Demazeau, Y., Vincent, J.: Evaluation of multi-agent systems: The case of interaction. In: *3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pp. 1–6. IEEE, Los Alamitos (2008)
7. Mylopoulos, J., Kolp, M., Giorgini, P.: Agent-oriented software development. *Methods and Applications of Artificial Intelligence*, pp. 748–748 (2002)
8. Sánchez-Pi, N., Fuentes, V., Carbó, J., Molina, J.M.: Knowledge-based system to define context in commercial applications. In: *Proc. of the 8th ACIS Conference SNPD 2007*, Tsingtao, China, pp. 694–699 (2007)
9. Wooldridge, M.J., Ciancarini, P.: Agent-oriented software engineering: The state of the art. In: Ciancarini, P., Wooldridge, M.J. (eds.) *AOSE 2000*. LNCS, vol. 1957, pp. 1–28. Springer, Heidelberg (2001)

Collaborative Workflow Management in Holonic Multi-Agent Systems

Fu-Shiung Hsieh

Department of Computer Science and Information Engineering,
Chaoyang University of Technology,
41349 Taichung County, Taiwan, R.O.C.
fshsieh@cyut.edu.tw

Abstract. Holonic multi-agent systems (HMS) provide a flexible architecture to achieve a goal based on dynamic organization and collaboration of autonomous agents. However, the structure of the holons in HMS presents a challenge in the design of an effective mechanism to dynamically organize agents and deliver the relevant information to the right agents at the right time to achieve a goal. How to provide a framework to realize effective human computer interaction in HMS is critical to the success and adoption of HMS. The objectives of this paper are to propose a viable design methodology to achieve effective human computer interaction in HMS based on context-aware computing technology and HMS architecture. Context-aware computing refers to an application's ability to adapt to changing circumstances and respond based on the context of use. A system is context-aware if it uses context to provide relevant information and/or services to the user at the point of need. To achieve these objectives, we first propose Petri net models to describe the workflows in HMS. Next, we propose models to capture resource activities in HMS. Finally, the interactions between workflows and resources are combined to obtain a complete model for workflow management. Based on the model, we propose architecture to dynamically generate context-aware user interface to guide the users and control resource allocation based on the state of HMS.

Keywords: Context aware, human computer interaction, workflow, holon.

1 Introduction

Many problem domains call for an agile and adaptive problem solving architecture to address the need for fast reaction to disturbances in real world. Multi-agent systems (MAS) are a paradigm to develop problem solver based on autonomous, co-operative and intelligent agents. A MAS consists of a collection of individual agents, each of which exhibits the characteristics of autonomy with respect to its actions and perception of a domain. Cooperation is achieved by autonomous computation within each agent and by communication among the agents. The capability of the resulting MAS is an emergent functionality that may surpass the capabilities of each individual agent. Agents consisting of sub-agents are called holonic agents or holons. The term "holon"

is based on the Greek word “holos” for “whole” and the suffix “-on” that denotes “part” [15]. In a holonic multi-agent system (HMS), a holon that appears as a single entity to the outside world may in fact be composed of many sub-agents and conversely, many sub-agents may decide that it is advantageous to join into the coherent structure of a super-agent and thus act as single entity. HMS provides a flexible and reconfigurable architecture to deal with changes based on dynamic organization and collaboration of autonomous entities called holons. However, the structure of the holons in HMS presents a challenge in design of effective mechanism to dynamically organize agents and deliver the relevant information to the right agents at the right time to achieve a goal. How to provide a framework to realize effective human computer interaction in HMS is critical to the success and adoption of HMS. Context-aware computing [1]-[8] is an effective technology to provide relevant information and/or services to the user at the point of need. The objectives of this paper are to propose a viable design methodology to achieve effective human computer interaction in HMS based on context-aware computing technologies and apply the methodology to a specific application domain.

Health care has been recognized as an important and promising field of context-aware research [16]. Workers in a real hospital are highly mobile; they change location and are assigned different tasks from time to time to perform their works. Furthermore, the role or mission of a hospital worker changes frequently. The information needed by these workers is highly dependent on their location, role, time, workflow and activity involved. With the wide acceptance of pervasive computing devices, how to streamline the processes by providing timely information to workers and exploiting the advantage of handheld devices is an important research issue. In this paper, we focus on application of HMS architecture and context-aware computing technology in the management of workflows in hospitals. A typical medical process is driven by the arrival of a patient’s appointment request. Depending on the type of appointment made by the patient, the medical workflow varies. Typical medical workflows require the formation of a collaborative network to handle a patient’s appointment request. How to effectively achieve collaborative network formation and take advantage of context-aware computing technologies to deliver the relevant information to hospital workers are important issues. The goal of this paper is to propose a framework to achieve effective human computer interactions. To achieve the objective, a model for representing workflow is required. In existing literature, one formal workflow knowledge representation and reasoning model is Petri net [8]. For examples, modeling and analysis of workflow processes have been extensively studied in [17]. Knowledge representation and reasoning based on Petri nets have also been studied in [9]-[11]. Our previous works [13], [14] on modeling of workflows and processes in multi-agent systems paves way for the development of this paper. In addition to representation of workflows, mechanism for agents to negotiate with one another must be developed. Contract net protocol (CNP) is a well known mechanism to distribute tasks in MAS. Application of CNP requires the development of a bid evaluation procedure specific to the problem [18], [19]. In [20], [21], [22], [23], an extension of contract net protocol has been proposed to achieve effective negotiation of agents in MAS. With the proposed Petri net models and the extended contract net

protocol, we propose a methodology in this paper to facilitate the design and analysis of context-aware workflow management applications in HMS.

The remainder of this paper is organized as follows. In Section 2, we introduce the workflow management problem. In Section 3, we propose resource activity models and workflow models in Petri nets. In Section 4, we propose a complete workflow management Petri net model and a resource allocation scheme. We conclude this paper in Section 5.

2 Workflow Management in HMS

In a health care system, a variety of resources are involved in the medical processes of patients. Different steps throughout the lifecycle of a medical process, from appointment reservation to discharge, require distinct resources for processing. A hospital’s staff is distributed in space or time, works in different shifts and their information needs are highly dependent on their location and other contextual conditions such as their role or the time of day. These characteristics pose challenges in the development of effective information systems and resource management scheme for hospitals. Due to the complexity of the tasks performed and the distributed information and resources, medical processes in a hospital require considerable mobility and coordination of resources. The above scenario exhibits a very complicated process flow and intensive messaging and interactions between different workers in the team. How to effectively manage the concurrent workflows as well as messages in the system is a challenge. To achieve effective management of resources, a process model is required.

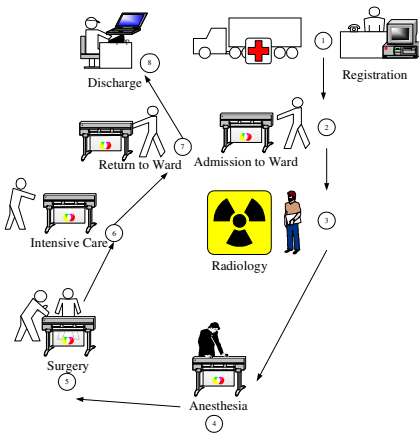


Fig. 1. A medical process

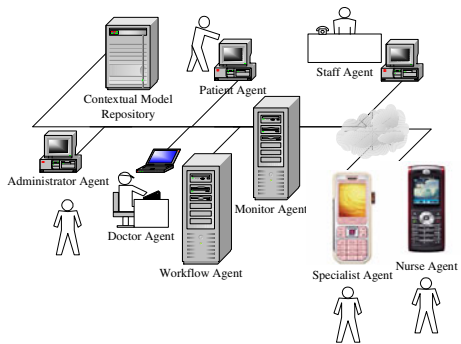


Fig. 2. System architecture

A medical process abstracts the procedures in a hospital. Each medical process consists of a set of tasks. Each task is associated with roles to be played by staff members. We are mainly concerned with the management of related resources for

effective enactment of these tasks. Each task has a unique identification, a description, and a set of roles that are capable of executing the task. Fig. 1 describes a typical surgery process. When a patient is scheduled for surgery after being admitted to the ward, he/she has to go through radiology tests. A surgery time is then assigned. Before the surgery, the head nurse initiates the procedures. She calls the ward, requests the ward officer to prepare the patient and calls the hospital porter to fetch the patient. The anesthesiologist is asked to go to the preparation room at a specific time. After the anesthesia, the patient enters the operating room with the surgeon and the assisting nurses. After finishing the operation, the patient will be moved to the intensive care room. If everything goes well, the patient will be returned to the ward.

We assume each worker is equipped with either a mobile handheld communication device such as Java phones or a computer used for communication with other team members. Although mobile phone provides a convenient media for communication, only short messages can be transferred via mobile phone due to its limited computing capabilities and display area. One of the design issues is to deliver the required information only to avoid overloading the mobile phone as well as the users. Another issue is to effectively enact the execution of operations to accomplish the task. Fig. 2 shows our proposed architecture for developing context-aware workflow management systems in HMS. In Fig. 2, the contextual model repository is used to capture the context of different types of team members and task workflows. There are six types of agents corresponding to different types of team members, including administrator agents, doctor agents, monitor agents, patient agents, staff agents, specialist agents and nurse agents. The information to be delivered to different members in the system varies depending on the role of the hospital team member involved in carrying out the task. Each workflow agent represents a specific workflow. The monitor agent aims to provide a model to monitor the progress of the medical processes. A team member usually needs to participate in different parts of the medical workflows. How to generate the context-aware information to guide a team member using a variety of different types of communication devices to collaboratively accomplish the task is a significant design issue.

3 Workflow Models and Capability Models of Agents

To facilitate automated generation of context-aware guidance information for each team member, an appropriate contextual modeling approach is adopted in this paper to capture the interactions between individual team members and different workflows. There are two requirements that must be met in the selection of appropriate models. First, the model must have well-established formal mechanisms for modeling as well as analysis. Second, the model must be accompanied with a standardized interchange format that can be shared among agents. Petri net [8] is a model that meets these two requirements. Therefore, we adopt Petri nets to model the contexts of agents. The advantages of applying Petri nets formalism to model and analysis of a context-aware system are summarized as follows. First of all, the graphical nature of Petri nets can visualize sequences of firing via token passing over the net. Secondly, Petri nets have

well-established formal mechanisms for modeling and property checking of systems with concurrent, synchronous and/or asynchronous structures. Thirdly, the mathematical foundation of Petri nets can analyze structural and dynamic behaviors of a system. These advantages make Petri nets a suitable modeling and analysis tool. Furthermore, the emerging Petri Net Markup Language (PNML) [12] is an XML-based interchange format for Petri nets. PNML makes it possible for individual agents or companies to exchange their process models in Petri net. In this paper, PNML is used for sharing and exchanging the process models of participants in a team.

A Petri Net (PN) [8] G is a five-tuple $G = (P, T, I, O, C, m_0)$, where P is a finite set of places, T is a finite set of transitions, $I \subseteq P \times T$ is a set of transition input arcs, $O \subseteq T \times P$ is a set of transition output arcs, $C : T \rightarrow S$ is a mapping that specifies the timing constraints, including the earliest firing time and the latest time, for firing each transition, where the set S is a finite set with each element $d \in S$ belonging to $R^+ \times R^+$ and $d_1 \cap d_2 = \Phi \forall d_1 \in S$ and $m_0 : P \rightarrow Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers. A marking of G is a vector $m \in Z^{|P|}$ that indicates the number of tokens in each place under a state. $\bullet t$ denotes the set of input places of transition t . A transition t is enabled and can be fired under m iff $m(p) \geq I(p, t) \quad \forall p \in \bullet t$. Firing a transition removes one token from each of its input places and adds one token to each of its output places. A marking m' is reachable from m if there exists a firing sequence s bringing m to m' . The reachability set $R(m_0)$ denotes all the markings reachable from m_0 . A Petri net $G = (P, T, I, O, C, m_0)$ is live if, no matter what marking has been reached from m_0 , it is possible to ultimately fire any transition of G by progressing through some further firing sequence.

For the medical process of Fig. 1, there are eight steps. Each step can be modeled using Petri nets.

To model a medical workflow as a Petri net, we use a place to represent a state in the workflow while a transition to represent an event or operation that brings the workflow from one state to another. The workflow of a medical process w_n is modeled by a subclass of Petri nets called a workflow net $w_n = (P_n, T_n, I_n, O_n, C_n, m_{n0})$, where the set P_n of places denotes the production states whereas the set T_n of transitions denotes the operations.

Definition 3.1: The workflow of a medical process w_n is a workflow net $w_n = (P_n, T_n, I_n, O_n, C_n, m_{n0})$. A Petri net $w_n = (P_n, T_n, I_n, O_n, C_n, m_{n0})$ is a workflow net [17] if and only if (i) w_n has a source place ε and a sink place θ with $\bullet \varepsilon = \Phi$ and $\theta \bullet = \Phi$ and (ii) if we add a new transition t to w_n to connect θ with ε , the resulting PN w_n is strongly connected. The augmented workflow net w'_n associated with w_n is the strongly connected PN obtained by add a new transition t to w_n to

connect θ with ε . As each transition represents a distinct operation in a task, $T_j \cap T_k = \Phi$ for $j \neq k$.

Fig. 3 shows ten workflows for the different steps in Fig. 1.

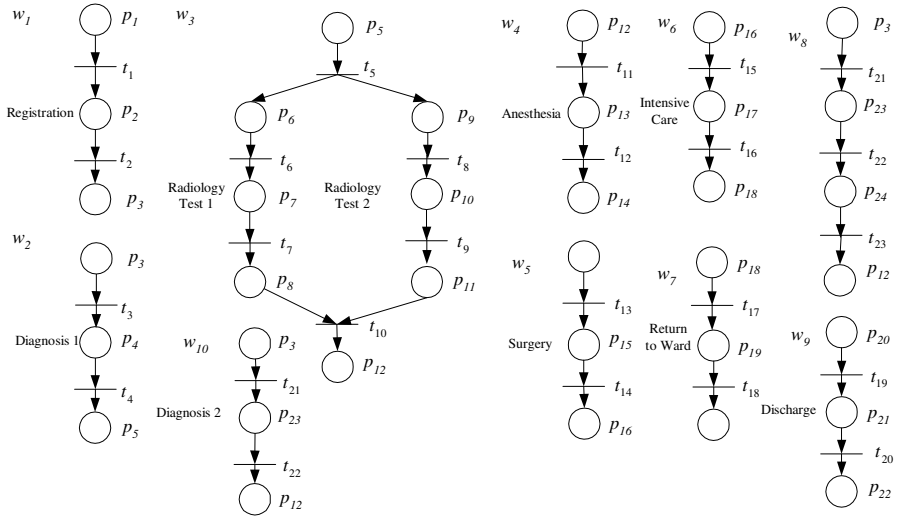


Fig. 3. Individual workflows $w_1 \sim w_8$

An activity is a sequence of operations to be performed and states to be visited by a certain type of agents (resources). We use a place in Petri net to represent a state of the agent activity. Each agent has an idle state. Each agent activity starts and ends with an idle state. An agent activity is described by a circuit in Petri net model. A circuit indicates that an agent activity includes resource allocation and de-allocation. Let R_n denote the set of agent types required to perform the operations in W_n . The Petri net model for the k -th activity of a type- r agent, where $r \in R_n$, is described by a Petri net A_r^k defined as follows.

Definition 3.2: Petri net $A_r^k = (P_r^k, T_r^k, I_r^k, O_r^k, C_r^k, m_r^k)$ denotes the k -th activity of a type- r agents, where $r \in R_n$. Remark that $T_r^k \cap T_r^{k'} = \Phi$ for $k \neq k'$.

Let K_r be the number of activities of a type- r agents. Let $\Omega_n^r \subseteq \{1,2,3,\dots, K_r\}$ denote the set of type- r activities in W_n . The initial marking m_r^k is determined based on the set of agents allocated to the k -th activity. More specifically, $m_r^k(p_r)$ is the number of agents allocated to place p_r , where p_r is the idle place of type- r agents.

Fig. 4 shows the activities of the seven types of agents involved in the workflows of Fig. 3.

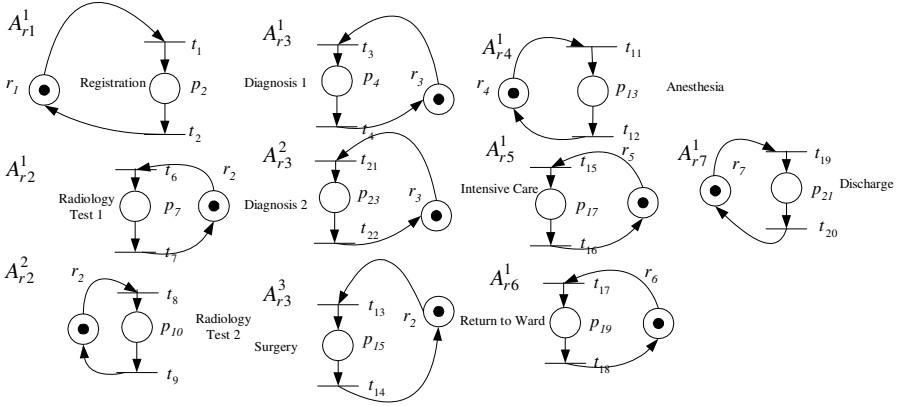


Fig. 4. Models of agent activities

4 Workflow Management

The monitor agent manages the progress of medical workflows based on a complete Petri net model. The complete Petri net model of a medical process is constructed by combining the agent activity models with the workflow model to take into account the interactions between agents and the workflow. Discovery of the required agent activity models for performing the operations of medical workflow is based on the extended contract net protocol proposed in [14]. To combine agent activity models with the workflow model, we define the operator “ \parallel ” as follows to merge two Petri net models with common transitions, places and/or arcs. By applying the “ \parallel ” operator, resources involved in an operation are synchronized so that the operation can be executed.

Definition 4.1: The complete Petri net model to process workflow W_n is modeled by $G_n = \parallel_{r \in R_n} A_r \parallel W_n$, where $A_r = \parallel_{k \in \Omega_r^r} A_r^k$.

Based on the composition operation defined in Definition 4.1, we may construct the complete Petri net model. Fig. 5 shows $G_n = W_n \parallel A_{r1}^1 \parallel A_{r2}^1 \parallel A_{r2}^2 \parallel A_{r3}^1 \parallel A_{r4}^1 \parallel A_{r5}^1 \parallel A_{r6}^1 \parallel A_{r7}^1$.

Property 4.1: Given G_n with marking $m \in R(m_0)$, there exists a control policy u such that G_n is live under m if and only if there exists a sequence of control actions that bring m to a marking $m' \in M_0$ with $m'(p_{r0}) \geq m_n^*(p_{r0}) \forall r \in R$, where $m_n^* \in M_n^*$ and M_n^* is the set of initial markings of G_n with minimal resources for the existence of a control policy to keep it live.

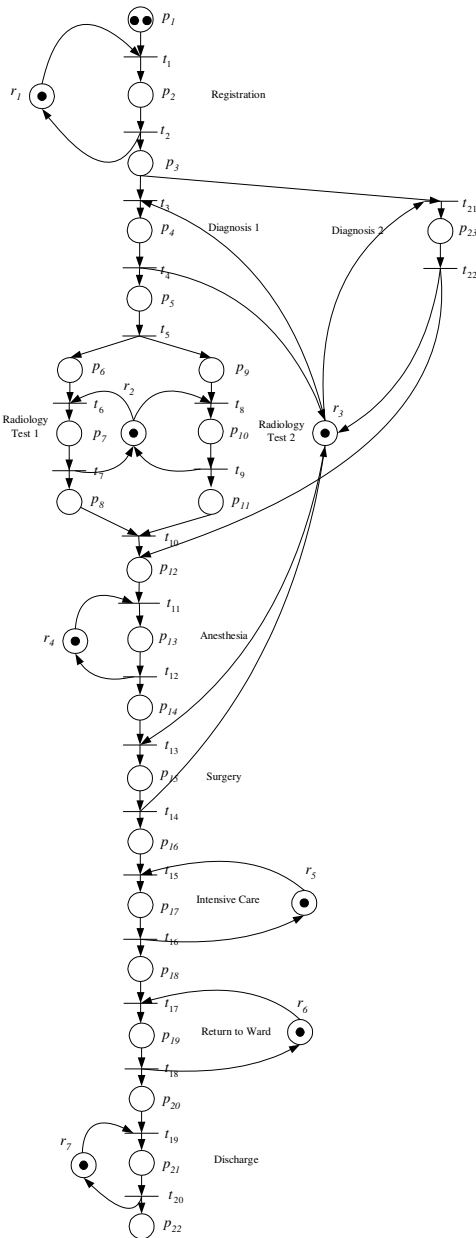
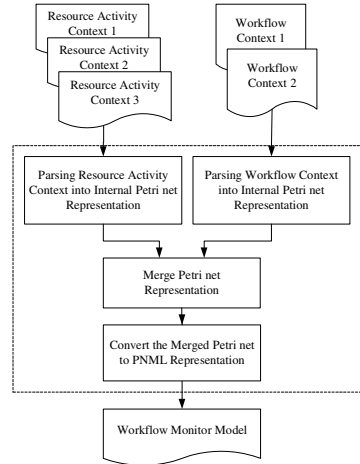
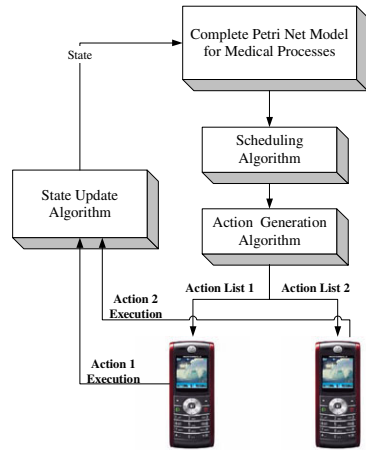


Fig. 5. A complete Petri net model



(a)



(b)

Fig. 6. Workflow management architecture

Application of Property 4.1 requires testing reachability of a marking m' that covers m_n^* . With the complete Petri net model, the state of the system can be captured, represented and predicted conveniently.



Fig. 7. Generated workflow driven action list

The model we propose previously can be used to monitor and control the resource allocation in a healthcare institution. Fig. 6 shows our proposed architecture for the implementation of a monitoring & resource allocation system based on our proposed Petri Net models. The complete Petri net model maintains the state of the medical processes based on the up-to-date state information from the state update algorithm. With the most up-to-date information of the medical processes, the scheduling algorithm sets the timing constraints for firing relevant transitions and allocates the required resources to satisfy the timing constraints. Finally, the action generation algorithm generates the action list for each worker. Different workers select their actions from the action lists for execution. By executing an action, a message is forwarded to the state update algorithm to update the state of the complete Petri Net model. The action generation algorithm then generates the action lists based on the new state. Suppose a token reaches place p_5 , which represents a decision needs to be made. In this case, our system will respond with the action list shown in Fig. 7 to instruct the user through the agent to either perform radiology test 1 or radiology test2. Fig. 8 shows the response of the system if radiology test1 is selected by the user.



Fig. 8. The response of the system

5 Conclusion

This paper focuses on the development of a novel approach to realizing context-aware workflow management in holonic multi-agent systems. We first propose workflow models in Petri net to describe the medical processes. Next, we propose resource activity models to capture the capabilities of agents and interactions between health care workers and resources and construct a complete Petri net model to capture the interactions between resources and medical processes. The complete Petri net models serve to control the allocation of resources. We develop a context-aware application to generate the action list to guide users involved in the medical processes.

Acknowledgement

This paper is supported in part by the National Science Council of Taiwan under Grant NSC97-2410-H-324-017-MY3.

References

1. Cheverst, K., et al.: Experiences of developing and deploying a context-aware tourist guide: The GUIDE project. In: 6th International Conference on Mobile Computing and Networking, Boston, pp. 20–31 (August 2000)
2. Patel, S.N., Abowd, G.D.: The ContextCam: Automated Point of Capture Video Annotation. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) *UbiComp 2004*. LNCS, vol. 3205, pp. 301–318. Springer, Heidelberg (2004)

3. Shi, Y., et al.: The smart classroom: Merging technologies for seamless tele-education. *IEEE Pervasive Computing* 2(2), 47–55 (2003)
4. Chen, H., et al.: Intelligent agents meet semantic web in a smart meeting room. In: 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 854–861 (July 2004)
5. Helal, S., et al.: Enabling location-aware pervasive computing applications for the elderly. In: 1st IEEE Conference on Pervasive Computing and Communications (March 2003)
6. Davis, F.D., Venkatesh, V.: Measuring user acceptance of emerging information technologies: An assessment of possible method biases. In: Proc. 28th Hawaii Int. Conf. System Sciences, pp. 729–736 (1995)
7. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. *IEEE Network* 8(5), 22–32 (1994)
8. Murata, T.: Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE* 77(4), 541–580 (1989)
9. Yu, S., Hsu, W., Pung, H.K.: KPN: a Petri net model for general knowledge representation and reasoning. In: 1998 IEEE International Conference on System, Man and Cybernetics, pp. 184–189 (1998)
10. Lianxiang, J., Huawang, L., Genqing, Y., Qingrong, Y.: An Improved Fault Petri Net for Knowledge Representation. In: *Proceedings of the 2009 Computational Intelligence and Software Engineering*, pp. 1–4 (2009)
11. Zhang, Z., Yang, Z., Liu, Q.: Modeling Knowledge Flow Using Petri Net. In: 2008 International Symposium on Knowledge Acquisition and Modeling, pp. 142–146 (2008)
12. Weber, M., Kindler, E.: The Petri Net Markup Language (2002), http://www2.informatik.hu-berlin.de/top/pnml/download/about/PNML_LNCS.pdf
13. Hsieh, F.-S.: Holarchy Formation and Optimization in Holonic Manufacturing Systems with Contract net. *Automatica* 44(4), 959–970 (2008)
14. Hsieh, F.-S., Chiang, C.Y.: Collaborative composition of processes in holonic manufacturing systems. *Computer In Industry* 62(1), 51–64 (2011)
15. Koestler, A.: *The Ghost in the Machine*. Hutchinson, London (1967)
16. Bricon-Souf, N., Newman, C.R.: Context awareness in health care: A review. *International Journal of Medical Informatics* 76(1), 2–12 (2007)
17. van der Aalst, W.M.P.: The application of Petri nets to workflow management. *J. Circuit. Syst. Comput.* 81, 21–66 (1998)
18. Hsieh, F.-S.: Model and Control Holonic Manufacturing Systems Based On Fusion Of Contract Nets and Petri Nets. *Automatica* (40), 51–57 (2004)
19. Hsieh, F.-S.: Analysis of contract net in multi-agent systems. *Automatica* 42(5), 733–740 (2006)
20. Hsieh, F.-S.: Collaborative reconfiguration mechanism for holonic manufacturing systems. *Automatica* 45(11), 2563–2569 (2009)
21. Hsieh, F.-S.: Dynamic composition of holonic processes to satisfy timing constraints with minimal costs. *Engineering Applications of Artificial Intelligence* 22(7), 1117–1126 (2009)
22. Hsieh, F.-S.: Developing cooperation mechanism for multi-agent systems with Petri nets. *Engineering Applications of Artificial Intelligence* 22(4-5), 616–627 (2009)
23. Hsieh, F.-S.: Design of Reconfiguration Mechanism for Holonic Manufacturing Systems based on Formal Models. *Engineering Applications of Artificial Intelligence* 23(7), 1187–1199 (2010)

Agent-Oriented Requirements Analysis from Scenarios

Lin Liu¹, Zhi Jin², Ruqian Lu², and Hongji Yang³

¹ School of Software, Tsinghua University, Beijing, China

²Academy of Mathematics and System Sciences, Chinese Academy of Sciences, China

³Software Technology Research Laboratory, De Montfort University, Leicester, England
linliu@tsinghua.edu.cn, zhijin@amss.ac.cn, rqlu@math.ac.cn,
hyang@dmu.ac.uk

Abstract. This paper proposes a scenario-driven agent-oriented requirement analysis methodology, where scenarios are textual descriptions of the interactions between various agents described by end-users. These scenarios are then transformed into an internal representation - Scenario-Trees. An inductive learning procedure is designed to decompose, cluster, and generalise the scenario descriptions to obtain an abstract grammar - an attribute grammar. The attributes and attribute computing rules are used to reinforce the expressiveness of the grammar. The various possible types of agents are analysed, the patterns of various agent types are defined. An experimental system SSAS, which generates Agent-Z specification for system from the original scenarios is implemented.

Keywords: software agent, requirements analysis, scenario, formal specification, inductive grammar learning.

1 Introduction

In recent years, there are many requirements engineering methodologies being extensively studied. One of the most popular approaches is object-oriented software analysis methods [1, 2] building upon the concept of objects, classes, properties and services, which considers that systems are made up of objects, interacting with each other through message passing. However, the OO analysis is greatly a design model, with which it is difficult to separate problem understanding from the implementation details. Requirements engineering methods requires a conceptual framework, whose concepts reflect the world view of the problem-solving perspective of the analysis methods. While several key requirements methods include three fundamental concepts: goal[3, 4, 5], agent[5, 6, 7], and scenario[8, 9, 10, 11], different approaches define the concepts of modeling and analysis differently, these three modeling elements stands for three essential modelling elements to form the basis for integrated requirements models.

Agents refer to entities residing in a particular environment with independent capacity and high level interaction with the others. Agent technology adopts the abstract concept of agent for system analysis, design, construction, and deployment. Because it provides a higher-level of modelling abstraction, it becomes a natural means to

control the complexity of system, and simply system development. Especially software agents are able to adapt to dynamic and open environments, research on this technology in many mainstream computer research fields are of concern and attention in recent years. In the 90's, the agent concepts and techniques are introduced field of requirements engineering [6], as the main concepts used to describe requirements modeling active entities in the external environment and within the system. Agent is not only a technology-oriented concept, but also a way of thinking. Agent-oriented analysis emphasizes using personification to the objective world, considers the problem space as an agent society, in which agent shows characteristics of autonomy, concurrency, who can collaborate to provide services to users. Agent-oriented requirements analysis method resolves many of the difficult problems that object-oriented analysis (OOA) method encounter when describing concurrent systems, intelligent systems and systems with uncertainty. First of all, the concept of the object is extended to agent concept, allows the previously separated notion of object and subsystem unified into the agent concept naturally. Second, in the scenarios setting, the system and users are studied in terms of their interactions, which is easy to understand, and easy to elicit the original requirements. In addition, the agent, goals and scenario integrated analysis focuses on the system functionality and behavior. Finally, the textual description-based methods facilitate the introduction of formal methods, thereby enhancing the accuracy of the results of the analysis.

2 Agent-Goal-Scenario Integrated Requirement Analysis

This article intended to provide a software requirement description tool for systems designers (including the end user) that transform informal pseudo-natural language description of the software system requirements documents into a formal description. Namely, an agent-goal-scenario integrated requirement analysis approach, AGSIRA, puts agent at the centre of the requirements analysis. First of all, users provide pseudo-natural language scenarios, each of which describes a series of interactions between external entities and system run-time entities. Then, an internal representation of the specification (Scenario Tree) is constructed based on the analysis to the pseudo-natural language textual descriptions, which is machine understandable. And then, we conduct inductive learning to the grammars, and derive an abstract grammar called system scenario grammar, which is the system requirements specification. Finally, the grammar is transformed to describe agent protocol in Agent-Z [13, 14, 15], provide the conceptual models of agent systems in the future.

2.1 AGSIRA Concepts

In AGSIRA, there are four basic data sets: N , V , A , G , represents sets of names, values, actions, goals, respectively.

Definition 2.1. Attribute t is defined as a two tuple $\langle n, v \rangle$, $n \in N$ represents the attribute name, $v \subseteq Value$ represents attribute value range, the set of attributes is denoted as At .

Definition 2.2. Entity e is defined as a subset of attribute $T \subseteq At$, and $T \neq \emptyset$, entity name e is the universal identifier of an entity, $e \in N$, set of attributes of entity e is denoted as $e.T$. The set of entities is denoted as E .

Definition 2.3. Agent a is defined as a three tuple $\langle T, C, G \rangle$, an agent is identified with its name, $a \in Name$, where $T \subseteq Attribute$, $C \subseteq Action$, $G \subseteq Goal$, while $T \neq \emptyset$, $C \neq \emptyset$, the set of actions of an agent a is denoted as $a.C$, the set of goals of an agent is denoted as $a.G$, the set of agent is denoted as Ag . If $a \in Agent$ and $a = \langle T, C, G \rangle$, when $a.G = \emptyset$, a is called a passive **agent**, also an object, the set of passive agent is denoted as Obj . when $a.G \neq \emptyset$, a is called an **agent**.

Definition 2.4. For any system S , there is a unique set of goals associated with G_s associated with S , If $a \in Agent$, and $a = \langle T, C, G \rangle$, when $a.G \subseteq G_s$, a is called an internal agent, or a system **agent**.

- (1) If $a \in Agent$, and $a = \langle T, C, G \rangle$, when $a.G \not\subseteq G_s$, it is called an external **agent**, or an **actor**.

Definition 2.6. An **Action-flow** is defined as follows :

- (1) $\forall ag \in Agent$, if $ag = \langle At, Ca, G \rangle$, $ca \in ag.Ca$, then ca is an action flow;
 (2) $\forall x, \forall y$, if x, y are action flows, then the sequential composition of x, y (denoted as $x \bullet y$) is also an action flow;
 (3) $\forall x, \forall y$, if x, y are action flows, then the parallel composition of x, y (denoted as $x \parallel y$) is also an action flow.

The set of action flow is denoted as AF .

Definition 2.7. **Scenario** is an action flow of several internal and external agents with common goal.

Definition 2.8. **Scenario-case** is a concrete execution trace of a scenario, which does not include fork, join, or selection.

Definition 2.9. **Context information** is the context related information, $Ct = \langle attributes, state-changes, constraints, conceptDef, Security-rules \rangle$, where *attributes* refers to the state of the world at the moment of analysis, *state-changes* refers to how the state of the world changed due to the execution of the given scenario, *Constraints* refers to the success and failure conditions of the current scenario, *ConceptDef* defines all relevant concepts, while *Security-rules* refers to system security policies.

3 From Textual Scenario Scripts to Structured Specification

Scenarios indicate the behavioral characteristics of a system when used in certain circumstances. It is important in requirements analysis since it is easy to elicit requirements in terms of concrete usage scripts. Based on the set of scenarios provided by user, we can identify key functions, usage mode, constraints, and derive the requirements specification based on it. These scenarios can also be used to validate the requirements, identifying inconsistencies and errors. This paper proposes to elicit

requirements based on scenario descriptions, and conduct inductive learning to scenario scripts, so that we can derive what system roles that the agent shall play, what are the obligations, what actions to take to fulfill these obligations, and what is the required knowledge. Once such information is obtained, problem can be understood in multiple levels of abstraction. Scenario Specification is the final product of requirement analysis stage. The external scenario specification uses key-word based pseudo-natural language description[16], it is provided to system developer or final user for documenting requirements, while there is also an internal representation, based on context-free language, CFL, provides the raw data for later automated requirements analysis activities based on grammar learning, including: concepts to be learnt, given scenario cases, and context information. The difference between the internal representation and external representation is: external representation mainly achieves user friendliness, so it is trying to mimic the most natural way of expression as people do in daily life. On the other hand, the internal representation aims to be more structured, and machine understandable. The intended expressiveness of the two languages is the same.

3.1 External Representation of Scenarios--SSDL

It is well understood that natural language is the most expressive representation of requirements knowledge. However, due to its inherent ambiguity, due to the different knowledge background, mindset and terminality, users and software development may diverge even facing the same requirements document. Formal languages are more rigor representations, but are not easily accessible to non-computer experts. This paper proposes a key-word based pseudo natural language - SSDL(Scenario-based Specification Description Language), as an instrument for users to describe their intended scenarios, as the input to the requirements process of AGSIRA. When designing SSDL, user friendliness is addressed by the careful selection of keywords and sentence structures, so that when final user describes their scenarios, the experience is close to writing natural language paragraphs with certain template or guidelines.

3.2 Internal Representation of Scenarios-- ScenarioTree

While SSDL is mainly for final user, we also need a more structured requirements representation for automated processing. The internal representation includes two parts: scenario grammar and scenario attributes. Scenario grammar includes CFL sentences, based on which, a CFG covering all existing scenarios will be generated and serve as the framework of requirements specification. The semantics of scenarios are mainly provided by the context information, which defines system boundary, input and output objects, state transitions and environment constraints. It provides the concrete attribute values of primitive concepts, and the values of intermediate variables in the grammar, which enables us to derive the sets and rules for attributes. The process of translate the external representation to internal representation include following steps: (1) separate grammar and attribute; (2) conduct the grammar learning process; (3) assign attributes to grammar rules.

4 Requirements Specification -- Scenario Grammar

Once the external scenario descriptions are translated into the internal scenario tree, we will get a domain specific attribute grammar, which can be used as the requirements specification, we called it a scenario grammar. Attribute grammar was used in compiling programming languages earlier, and was applied in several domains, including syntax-driven editor and environments, etc. It has become an effective instrument in software research. By associating semantic rules to a given CFG, the static semantics of a given language can be defined. The semantic rules are computation rules for attributes, and define the dependency relations between attributes. The definition for the scenario grammar is as follows:

Definition 4.1. Scenario Grammar(SG) are defined as a five tuple: $SG = \langle N, T, P, S, A \rangle$, where N is the set of non-terminals, T is the set of terminals with attributes, P is the set of attribute computation production rules, $S \in N$ is name of systems, which is the initial symbol, A is the set of attributes.

$N \cap T = \emptyset$, $V = N \cup T$ is the symbols list of SG, T^* is the set of terminals (including the set of null string, λ), V^* is the set of terminals and non-terminals. $V^+ = V^* \setminus \{\lambda\}$. The language generated with SG, $L(SG) \subseteq T^*$. The production rules in P are in one of the following format: $X \rightarrow \alpha$, $X \in N$, $\alpha \in V^*$.

Definition 4.2. Scenario Grammar Production Rules in P are:

$S \rightarrow S \parallel S$	/*parallel composition rule*/
$S \rightarrow S ; S$	/*sequential composition rule*/
$S \rightarrow Na$	/*atomic rule*/
$Na \rightarrow \text{action} - Ob$	/*internal action rule*/
$Na \rightarrow \text{action} - Ob - \text{channel}$	/*communication action rule*/
$Ob \rightarrow Ob, Ob$	/**/
$Ob \rightarrow ob$	

5 Multi-agent System Specification Language -- Agent-Z

Agent-Z is an agent-oriented extension to the Z language: (1) define all system entities as agent, thus define Agent class. It is the super-class of all concrete agent classes. There are instances of the concrete agent classes acting in the system, with different attribute and behavior. (2) While using the Z language with the extended Agent class to describe system structure, introduce process algebra to describe the communication activities of system.

Agent-Z has following grammar rules: (1) each input or output operation is associated with a communication channel. When declaring an input or output variable, it has to declare through which channel it is sending or receiving messages. (2) each operation in an agent class has to define its execution thread in the predication part of schema, and the operation name is referred to as *op* by default. (3) each agent class has a life thread definition in its predication part, referred to as *LP*. Since agent are active inherently, once it is created, it will be activated, start its life cycle determined

by the definition of its life thread. (4) The dynamic relationships among agents are defined in the predicate part of the specification, e.g. parallel, role evolvement etc.

Here is a definition to the percept operation of a reactive agent controlling the environment temperature:

```

-----Percept-----
 $\Delta t$ 
req-chan: Channel
req-chan? $t_c$ : Temperature
inform-chan! inform: BOOLEAN
-----
 $t' = t_c$ 
 $t_{min} < t' < t_{max} \Rightarrow inform = true$ 
 $t_{min} > t', t' > t_{max} \Rightarrow inform = false$ 
Percept = req-chan?  $t_c$  ; inform-chan ! inform ; Percept
-----
    
```

The predication part of operation Percept in the lower part of the schema fragment defined the operation thread in the last sentence, the left side of the equal sign is the name of current operation, while the definition on the right hand side says: we first wait for an input from the req-chan channel, whose type is Temperature, whose value will be stored in the variable t_c . When t_c is normal, the value of inform is true, when t_c is abnormal, the value of inform is false, we then send a warning to inform-chan channel. When Agent executes percept action, it will repeat the actions above. The purpose of defining action execution thread is to describe agent communication processes during system execution.

● Life process definition for Agents

The fundamental difference between Agent class and generic object class is that it has a life process, defining the behavior of an agent in its life cycle. Once a system agent is created, it will be assigned a control thread, executes the life thread in parallel with other agents. Here, the way to substitute agent class name and operation name is: identify the definition of the agent or the operation in the specification, and substitute it with the life thread definition or operation thread. Here is a definition to the reactive agent controlling environment temperature in Agent-Z:

```

-----TemperatureDetector-----
inform-chan: Channel /***** */
 $t_{max}$ :Temperature /* constant definition */
 $t_{min}$ :Temperature /***** */
/*Here is the state definition for agent class TemperatureDetector*/
-----
| t:Temperature;
|  $t_0$ :Temperature;7
-----
/* Here is the definition of an operation of agent class TemperatureDetector */
-----Percept-----
 $\Delta(t, t_0)$ 
req-chan: Channel
req-chan? $t_c$ : Temperature
inform-chan! inform: BOOLEAN
    
```

```

|-----|
|  $t_0' = t$  |
|  $t' = t_c$  |
|  $t_{min} < t' < t_{max} \Rightarrow inform=true$  |
|  $t_{min} > t', t' > t_{max} \Rightarrow inform=false$  |
|  $op = req-chan? t_c. inform-chan ! inform . op$  |
|-----|
|/*Here is the definition of an operation of agent class TemperatureDetector */|
|-----|
| React |
|  $\Xi(t, t_0)$  |
|  $inform-chan?inform: BOOLEAN$  |
|  $switch-chan!switch: Switch$  |
|-----|
|  $inform=false \wedge t < t_{min} \wedge t_0 > t_{min} \Rightarrow switch=tune-up$  |
|  $inform=false \wedge t > t_{max} \wedge t_0 < t_{max} \Rightarrow switch=tune-down$  |
|  $op = inform-chan?inform . [inform=false] switch-chan! switch . op$  |
|-----|
|/* Here is the life thread definition for agent class TemperatureDetector */|
|/* instances of TemperatureDetector executes Percept and React in their lifecycle*/|
|  $TemperatureDetector \triangleq (Percept || React)$  |
|-----|

```

In multi-agent systems, all running entities are agents, so the concepts of operation and agent are fairly similar, and transformable. E.g. the operations *Percept* and *React* can be defined as two agents as well. Agent class *TemperatureDetector* is an agent class inheriting *Percept* and *React*. From the definition of operation thread and life thread, we can see the difference between the two is that in life thread, any previously defined operation name can be used, while in operation thread, only three basic action definitions can be used, namely, input ($x?y$), output ($x!y$), and silent(τ).

● Bisimulation in Agent-Z

Bisimulation is the major relation considered between processes in process algebra[17,18]. It describes the equivalence of communication behavior between processes. We use it as one of the basic relational operator in Agent-Z. It is denoted with a “ \sim ”. We can use it to analyse and refine the communication relationships between agents. For example, in the temperature control system example above, we assume that there is a process P, the iterative execution of P will yield $!P$ (the $!$ in front of P is iteration operator), $!P$ bisimulates agent *Percept*, then P bisimulates $req-chan?t . inform-chan! f(t)$, represented in(1)

$$\exists P \bullet Percept \sim !P \Rightarrow P \sim req-chan?t . inform-chan! f(t) \quad (1)$$

At the same time, assume that there is a process Q, if we execute Q repetitively, we will get $!Q$, if $!Q$ bi-simulates agent *React*, the Q bisimulates $inform-chan?m . [h(m,t_0)]switch-chan! g(m,t_0)$, represented in(2)

$$\exists Q \bullet React \sim !Q \Rightarrow Q \sim inform-chan?m . [h(m,t_0)]switch-chan! g(m,t_0) \quad (2)$$

Similarly, if there is a process R, if R executes repeatedly, we get $!R$, if $!R$ bisimulates agent *TemperatureDetector*, then we can derive R bisimulates ($req-chan?t .$

$[h(m,t_0)]\text{switch-chan! } g(f(t), t_0)$, as in(3)

$$\begin{aligned} \exists R \bullet LP(\text{TemperatureDetector}) \sim !R \Rightarrow R \sim (v \text{inform-chan}) (req\text{-chan?}t . \text{inform-chan! } f(t) \\ \parallel \text{inform-chan?}m . [h(m,t_0)]\text{switch-chan! } g(m,t_0)) \Rightarrow R \sim (req\text{-chan?}t . \tau.0 \parallel [h(m,t_0)] \\ (\text{switch-chan! } g(f(t), t_0)) \Rightarrow R \sim (req\text{-chan?}t . [h(m,t_0)]\text{switch-chan! } g(f(t), t_0)) \end{aligned} \quad (3)$$

Based on the definition of agent TemperatureDetector in (1) (2)(3) , when we do the operation substitution to agent P:

$$\{switch\text{-chan} / \text{inform-chan}, g(f(t), t_0) / f(t)\} \quad (4)$$

Then, we can derive: $R \sim P$

Based on such a reasoning process, we find that from the perspective of communication relation, React operation is redundant, when we need to reduce communication overhead, we can substitute percept operation, and the function of react operation can be reassigned to percept, so that react operation can be omitted. Only when there are needs for distributions, we add react operation in the system, but it won't affect the system – environment communication.

● Combining temporal logic operator and bisimulation

In Agent-Z, the communication behavior is defined as :

$$\begin{aligned} ComAction ::= & \tau \mid simple \langle\langle Channel \times Direction \times Name \rangle\rangle \\ & \mid parallel \langle\langle ComAction \times ComAction \rangle\rangle \mid choice \langle\langle ComAction \times ComAction \rangle\rangle \\ & \mid sequence \langle\langle ComAction \times ComAction \rangle\rangle \\ Direction ::= & ? \mid ! \end{aligned}$$

Communication actor involves more than one agent, so we cannot assign it to any one agent. The treatment in this paper is: (1) define a pair of actions for each communication, the sender and the receiver, the agent attending the communication either plays sender or receiver. (2) There are always group agents in multi-agent systems, composed of two or more communicating agents. (3) For any given agent, if all agents that it is communicating are external agents, then it is an agent representing the entire system.

6 From Scenario Specification to Agent-Z Specification

6.1 Elicit Requirements from Scenario Specification

This paper proposes to conduct inductive learning to known scenarios, from which derive generic scenarios, and combine partial description into complete ones, then further extend and synthesis, to generate system requirements specification. The input of the elicitation is the external representation of scenarios, while the output is the requirements specification of the scenario grammar. The process including following 5 steps: scenario filtering, scenario decomposition, scenario classification and comparison, scenario extension and synthesis, specification refinement.

Definition 6.1. Scenario specification is composed of $m(m>0)$ scenarios, $sc_i(0<i\leq m)$, denoted Scen-Set(S), for each scenario, sc_i is a sentence of the language defined by

SG. System scenario grammar is an abstract grammar between generic scenario grammar and concrete scenarios, it covers all existing wanted scenarios, while exclude all unwanted scenarios.

There are production rules in the form of $S \rightarrow \alpha$ ($\alpha \in V^*$) in the production rule set P of system scenario grammar. For each non-terminal appears in α , there is an instance of Y, when substituting Y with this scenario, we can recover the scenarios of S.

If there is no non-terminals in α , then $X \rightarrow \alpha$ is called a common model of S(scenarios). The kind of models interested most in this paper are the ones with non-terminal (requirements specifications) denoted as: $\text{Model}(S) (\subset V^*)$.

After the filtering of the information above, the scenario specification only remains system agents for future system.

Definition 6.2. The first level decomposition of scenario $\xi \in \text{Scen-Set}(S)$ is : $S \rightarrow A_1 \parallel A_2 \parallel \dots \parallel A_n, n \geq 1$, where A_1, A_2, \dots, A_n is the named agents in A, $A_i \in V^+, i=1, \dots, n$.

6.2 From Requirements Specification to Agent System Specification

Now we need to generate the agent system specification based on the requirements specification obtained by grammar learning, here the agent system specification is represented with agent-Z. In order to implement this step, the major issue to be solved is to build a clear mapping between the elements in scenario grammar SG and the elements in Agent-Z, so that the scenario grammar can be transformed into Agent-Z. Based on the above analysis and our previous knowledge, there four types agents whose paradigm can be defined:

- Passive agent(object) : For this kind of agent, the first action is always waiting for input, as it needs to be triggered by external events. When representing in Agent-Z, the template is as such: passive agents are :

$agent \sim (x_1?y ; P_1 + \dots + x_n?y ; P_n); agent, P_1, \dots, P_n$ represents the action flow.

- Reactive agent : For this kind of agent, according to the difference environment status, different reactive actions will be executed, they are able to monitor the changes in environment. The behavior of reactive agent can be controlled with a set of “environment condition \rightarrow reaction” rules. In Agent-Z specification, reactive agents are :

$agent \sim (x!y; y?z; ([p_1(z)]P_1 + [p_2(z)]P_2 + \dots + [p_n(z)]P_n)); agent$

Among which, $pi(z)$ is a first order predicate including variable z, here the variable z can be a tuple.

When $i \neq j, p_i(z) \wedge p_j(z) = \emptyset, p_1(z) \vee p_2(z) \vee \dots \vee p_n(z) \equiv \text{true}$,

P_1, P_2, \dots, P_n are the set of potential reactions agent in current environment.

- **Deliberate agent** : For this kind of agent, there are internal inference engine, which can plan for future behavior based on environment and system status. It often embedded with learning ability. In Agent-Z, deliberate agents are :

agent $\sim (x!y . y?z . ([p_1(f_1(z,at))]P_1 + [p_2(f_n(z,at))]P_2 + \dots + [p_n(f_n(z,at))]P_n))$

among which, $p_i(f(z,at))$ is a first order predicate logic, if $i \neq j$, $p_i(f(z,at)) \wedge p_j(f(z,at)) = \emptyset$

$p_1(f_1(z,st)) \vee p_2(f_2(z,at)) \vee \dots \vee p_n(f_n(z,at)) \equiv \text{true}$,

at is the internal state of agent,

P_1, P_2, \dots, P_n are the plan of agen, in P_i , the current status of agent at may change, which led to changes in $f(z,st)$.

- **Social agent** : For this kind of agent, not only its own status need to be recorded, it also need to build knowledge model of other agents, to decide the next action, e.g., whether to collaborate with someone, who to collaborate with, whether to get into competition, using what strategy to compete, etc. In agent-Z, social agents are represented as :

agent $\sim (x!y . y?z . ([p_1(f_1(z,at,st))]P_1 + [p_2(f_2(z,at,st))]P_2 + \dots + [p_n(f_n(z,at,st))]P_n))$

Among which , $p_i(f_i(z,at,st))$ is first order predicate, when $i \neq j$, $p_i(f_i(z,at,st)) \wedge p_j(f_j(z,at,st)) = \emptyset$

$p_1(f_1(z,st)) \vee p_2(f_2(z,st)) \vee \dots \vee p_n(f_n(z,st)) \equiv \text{true}$, st is the internal state of agent.

Lemma 6.1. Passive agent, reactive agent, deliberate agent and social agent covers all possible agent types. In other words, all agent represented by Agent-Z specification falls into one category of the above agents.

Having defined the basic types of agents, we can further classify social agents into more specific types of agents, such as honest agent, loyal agent, altruism agent, profit-driven agent, etc.

Definition 6.3. When the scenario shows that when an agent receives an inquiry to his own status, he will reply with the actual status information, then the agent is defined as an honest agent. $S \vdash \forall o \in \text{Output}(a) \Rightarrow o \in \text{facts}(a)$.

Definition 6.4. When the scenario shows that when an agent receives an inquiry about his knowledge of a third party, he will not disclose the private information, then the agent is considered as a loyal agent.

$S \vdash \forall o \in \text{Output}(a) \Rightarrow o \notin \text{secrets}(a)$.

Definition 6.5. When the specification shows that when an agent is capable of providing a service to others, he will commit to it, then the agent is considered an altruism agent.

$S \vdash \forall r \in \text{request}(a) \wedge \text{capable}(a, r) \Rightarrow \text{commit}(a, r)$.

Definition 6.6. When a scenario specification shows that, an agent will only provide to others if there are profits, it is considered a profit driven agent.

$S \vdash (\forall r \in \text{request}(a) \wedge \text{capable}(a, r) \wedge \text{profit}(r) \Rightarrow \text{commit}(a, r))$

$\wedge (\forall r \in \text{request}(a) \wedge \text{capable}(a, r) \wedge \neg \text{profit}(r) \Rightarrow \neg \text{commit}(a, r))$.

7 Conclusions

As the existing scenario-based approaches serve different purposes, use different representational features, and have different analysis capabilities, the concept of scenario needs to be differentiated along the dimensions.

The combined use of agents, goals and scenarios have been explored within RE, primarily for eliciting, validating and documenting software requirements [19,20]. Van Lamsweerde and Willement studied the use of scenarios for requirement elicitation and explored the process of inferring formal specifications of goals and requirements from scenario descriptions in [4]. Though they thought goal elaboration and scenario elaboration are intertwined processes, their work regarding scenarios in [5] mainly focuses on goal elicitation. Our emphasis is the other way around, i.e., how to direct scenario –based design.

In the CREWS project, Collete Rolland et al. have looked into the coupling of goal and scenario in RE with CREWS-L'Ecritoire [8]. In CREWS-L'Ecritoire, Scenarios are used as a means to elicit requirements/goals of the system-to-be. Their method is semi-formal. Both goals and scenarios are represented with structured textual prose. The coupling of goal and scenario could be considered as a “tight” coupling, as goals and scenarios are structured into <Goal, Scenario> pairs, which are called “requirement chunks”. Their work focuses mainly on the elicitation of functional requirements/goals.

In AGSIRA, given textual descriptions (in natural language) for scenario models a semi-formal notation can be extracted during the early stages of requirements analysis, to help explore and prune the space of system behavior. The agent-orientation, goal-orientation and scenario-orientation perspective of this paper complement each other in requirements elicitation, representation and analysis, while will passed to the incremental system design process. The combined use of languages of different characteristic enables the description of functional and non-functional requirements at different abstraction levels, namely, high-level requirements and concrete system scenarios, intentional strategic rationales of agents, and non-intentional details of concurrent, temporal aspects of the agent system under development.

Acknowledgements

This paper is partially supported financially by the National Natural Science Foundation (Grant No. 60873064, 90818026) and National Basic Research Program of China (973) (Grant No. 2009CB320700).

References

1. Jacobson, I.: *Object-Oriented Software Engineering, A Use Case Driven Approach*. Addison-Wesley Publishing Company, New York (1992)
2. Iivari, J.: *Object-Orientation as Structural, Functional and Behavioural Modeling: A Comparison of six Methods for Object-Oriented Analysis*. *Information and System Technology* 37(3), 155–163 (1995)

3. Anton, A.I.: Goal-based Requirement Analysis. In: Proc. International Conference of Requirements Engineering (ICRE 1996), pp. 136–144. IEEE computer Soc. Press, Colorado Springs (1996)
4. Lamsweerde, A.V., Willemet, L.: Inferring Declarative Requirements Specifications from Operational Scenarios. IEEE Transactions on Software Engineering (December 1998); Special Issue on Scenario Management
5. Letier, E.: Reasoning about Agents in Goal-Oriented Requirements Engineering. PhD Thesis. Department of Information Engineering, Catholic University of Louvain (May 2001)
6. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE 1997), Washington D.C., USA, January 6–8, pp. 226–235 (1997)
7. Giunchiglia, F., Mylopoulos, J., Perini, A.: The Tropos Software Development Methodology: Processes, Models and Diagrams. In: Giunchiglia, F., Odell, J.J., Weiss, G. (eds.) AOSE 2002. LNCS, vol. 2585, pp. 162–173. Springer, Heidelberg (2003)
8. Rolland, C., Grosz, G., Kla, R.: Experience With Goal-Scenario Coupling In Requirements Engineering. In: Proceedings of the IEEE International Symposium on Requirements Engineering 1998, Limerick, Ireland (June 1999)
9. Sutcliffe, A.: Scenario-Based Requirement Analysis. Requirements Eng. 3, 48–65 (1998)
10. Haumer, P., Pohl, K., Weidenhaupt, K.: Requirement Elicitation and Validation with real World Scenes. IEEE Transactions on Software Engineering 24(12) (1998); Special Issue on Scenario Management
11. Hsia, P., Samuel, J., Gao, J., Kung, D.: Formal Approach to Scenario Analysis. IEEE Software (March 1994)
12. Rednell, B., Kimbler, Wesslen, A.: Improving the Use Case Driven Approach to Requirement Engineering. In: Proceeding RE 1995, York UK, pp. 40–47 (1995)
13. Bottaci, L., Jones, J.: Formal Specification Using Z: A Modelling Approach. Thomson Publishing (1995)
14. Duke, R., Rose, G., Smith, G.: Object-Z: A Specification Language Advocated for the Description of Standards. Technical Report No.94-45, Software Verification Research Centre, Department of Computer Science, The University of Queensland
15. Luck, M., d’Inverno, M.: A Formal Framework for Agency and Autonomy. In: Rao, A., Singh, M.P., Wooldridge, M.J. (eds.) ATAL 1997. LNCS, vol. 1365, pp. 49–63. Springer, Heidelberg (1998)
16. Lu, R., Jin, Z., Wan, R.: Requirement Specification in Pseudo-Natural Language in PROMIS. In: Proc. of 19th COMPSAC, pp. 96–101 (1995)
17. Milner, R., Parrow, J., Walker, D.: A Calculus of mobile Processes, Parts 1-2. Information and Computation 100(1), 1–77 (1992)
18. Hennessy, M., Rathke, J.: Bisimulations for a calculus of broadcasting systems. Theoretical Computer Science 200, 225–260 (1998)
19. Fisher, M., Wooldridge, M.: “On the Formal Specification and Verification of Multiagent systems. International Journal of Cooperative Information Systems 6(1) (1997)
20. Hong, Z.: A formal specification language for agent-oriented software engineering. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) AAMAS 2000 and AAMAS 2002. LNCS (LNAI), vol. 2636, Springer, Heidelberg (2003)

Application of Multi-Agents in TV Commercial Recognition System

Pawel Biernacki

University of Technology Wroclaw, Janiszewskiego 7/9, 50-350 Wroclaw, Poland
pawel.biernacki@pwr.wroc.pl

Abstract. A Multi-agent system (MAS) which is a collection of agents cooperating with each other in order to fulfill common and individual goals can be used to implement a multimedia system for TV commercial identification. In this paper the effective autonomous method of a single commercial extracting from a advertising block and its recognition using only the audio signal based on MAS model is presented. Proposed solution uses a multidimensional orthogonal audio signal representation for a track parametrization and gives a recognition at the level of 98%.

Keywords: commercial recognition, signal parametrization, multi-agent system.

1 Introduction

One of the main function of the media research companies is radio and television commercial blocks monitoring. To investigate what and when is broadcast the efficient and intelligent methods of a single commercial extraction and recognition are needed.

Recently, a more intelligent systems has been introduced in form of the MAS that can act in the place of a human being. Proposed commercial recognition system is a kind of intelligent pattern classification agent system. The conceptual model of such system is as follow [1]. The Intelligent Pattern Classification Agent System consists of main agent and the Management Agent that has overall control over the other agents (pattern recognition agents) [2]. In [3] the features extraction system is proposed which is as follows: Feature Definition Module for feature and relation definition, A Module in which primitive routines for data processing and feature extraction are gathered, Feature Management Agents (MA) for storing data and features, Feature Extraction Agents (EA) for feature extraction.

Existing solutions for features extraction from audio and video signals base on Hidden Marcov Models [5], video signals [6] [7] or use different frequency domain 'audio fingerprinting' methods [10] [12] [11]. The aim of the study is to propose method for identifying advertising spots, which may differ from each other only small piece, such as a single word. In addition, the new method should work well for signals of poor quality recording. Every day about 10000 commercials

are broadcasting, which must be recognized. Therefore, improving the quality of diagnosis, even by 1% compared with existing methods is a big achievement.

This article consists of the following parts. In section 2 the multi-agent system architecture is discussed. Some simulation results of proposed solution are presented in section 3.

2 Proposed Multi-agent System Architecture

The multi-agent system is composed of five main components: management agent, advertising block detection agent, single commercial extraction agent, features extraction agent, database searching agent (Fig. 1).

2.1 Management Agent (MA)

This agent has overall control over the other agents. If it receives a message from another agent, it can response by selecting a suitable agent for the message and activates this agent. MA orders the action with different parameters (commercial detection, features extraction). It decides what to do with unrecognized commercials. Agent creates a report on the operation of the system, it creates a list of the identified advertisements as well as those new to the overwritten database.

2.2 Advertising Block Detection Agent (ABDA)

The advertising block detection agent operates on a TV broadcasting signal. It uses video signal and audio signal to find the start and the end moments of the advertising block in TV program. It is done by verification TV station logo presence (it disappears during commercials) and advertising block distinctive jingle at the start and end moment of commercials. Detection of the block is reported to the management agent that activates the single commercial extraction agent.

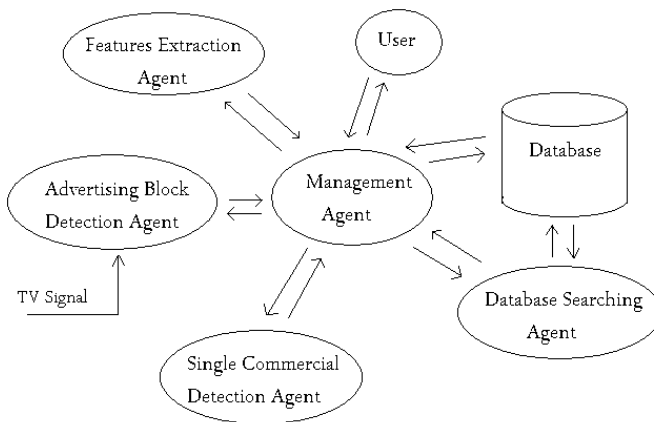


Fig. 1. Multi-agent commercial recognition system

2.3 Single Commercial Extraction Agent (SCEA)

This agent using only audio signal divides advertising block into individual ads. For the commercial start and end points the following elements can be used:

- short silence period after each commercial (no every silence denotes commercial end)
- spectrum differences before and after silence period
- commercial duration is often a multiple of 5 seconds periods

Listed above conditions for start/end commercial point identification have practical applications in TV commercial block division. In the radio broadcasting case the commercials often don't have clear start/end points (e.g. a track or station identifier appears in background). In such situations a single commercial separation is very difficult or even impossible.

Proposed intelligent solution consists of the following three steps:

- **step 1** - silence detection; empirical research shows that silence duration between commercials is from 60 ms to 200 ms long. Different silence periods are very often the parts of the commercials. To detect silence period the two criteria are used:
 - audio signal $x(t)$ magnitudes are between some maximum value

$$|x(t)| < Max, \quad Max - const \quad (1)$$

- signal energy is below some constant value

$$Energy = \sum_{t=t1}^{t2} |x(t)|^2 < Energy_{max} \quad (2)$$

where $t1$ and $t2$ are start and end points when condition (1) is true.

- **step 2** - power spectrum density (Welch method) of the audio signal fragments of the duration $T1[s]$ ($T1$ - algorithm parameter) before and after silence period are calculated. If the distance between the spectrum is too big (there is a small correlation between examined signal fragments) then the detected silence period (step 1) is a potential start or end point of the commercial
- **step 3** - having the set of potential start/ end points of the commercials and assuming the commercial duration as a multiple of 5 seconds periods the proper start points of the following commercials are established (Experimental results allowed to estimate the probability density function of appearing the commercial of the particular duration. This function is used to take a decision of the start/end point appearing).

Individual commercial extraction event is reported to the management agent. It checks whether the features extraction agent is free. If so, it allows the SCEA to send a single ad for its parameterization.

2.4 Features Extraction Agent (FEA)

In proposed solution a single commercial is parameterized, making up 'the commercial fingerprint'. FEA uses a multidimensional orthogonal audio signal representation for a track parameterization. Using the results presented in [8], the following multi-dimensional Fourier series expansion of the commercial audio signal (x) is obtained

$$\begin{aligned} \hat{x}_0 = & \sum_{i_1=0}^N x \rho^{i_1} x_0^{i_1} + \sum_{i_1=0}^N \sum_{i_2=i_1}^N x \rho^{i_1, i_2} x_0^{i_1, i_2} + \dots \\ & \dots + \sum_{i_1=0}^N \sum_{i_2=i_1}^N \dots \sum_{i_M=i_{M-1}}^N x \rho^{i_1, \dots, i_M} x_0^{i_1, \dots, i_M} \end{aligned} \tag{3}$$

where

$$\begin{aligned} x \rho^{i_1} &= (x_0, r_0^{i_1}) \\ x \rho^{i_1, i_2} &= (x_0, r_0^{i_1, i_2}) \\ &\dots \\ x \rho^{i_1, \dots, i_M} &= (x_0, r_0^{i_1, \dots, i_M}) \end{aligned} \tag{4}$$

are the generalized (multi-dimensional) Fourier [9] (i.e. Schur-type) coefficients. The coefficients (4) can be interpreted as the orthogonal representation of the commercial audio signal (commercial track) in the multidimensional space.

The diagram of the FEA parameterization functionality is presented in Figure 2.

The Schur coefficients (4) can be used for audio signal parameterization. Entire audio commercial parameterization procedure can be done in the tree steps:

1. dividing audio commercial signal into the one second long pieces

$$y_{T_k} = \{y_T(n); n = k * f_P, k * F_p + 1, \dots, k * 2 * f_p - 1\} \tag{5}$$

where k means k -th signal section and $k = 0, 2, \dots, K - 1$.

2. computing Schur coefficients (4) vector for each piece
3. construct ID matrix for the audio commercial (using Schur coefficients vectors as its rows)

$$YM = \begin{pmatrix} \rho_{0;T_0}^0 & \dots & \rho_{0;T_1}^{j_1, \dots, j_w} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{0;T_{K-1}} & \dots & \rho_{0;T_K}^{j_1, \dots, j_w} & \dots \end{pmatrix} \tag{6}$$

Commercial 'fingerprint' extraction is reported to the management agent. It checks whether the database searching agent is free. If so, it allows the FEA to send the 'fingerprint' to its identification.

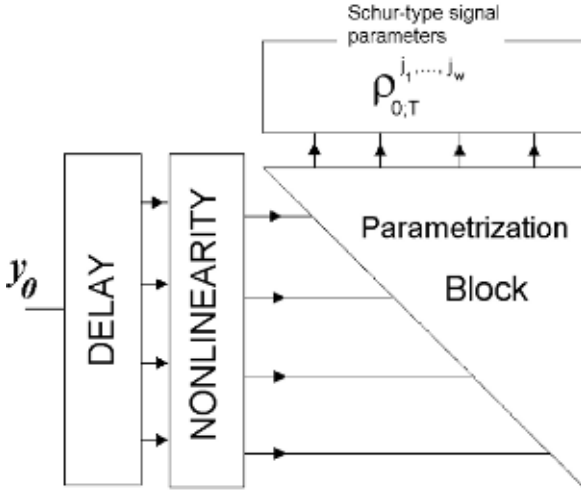


Fig. 2. FEA parameterization diagram

2.5 Database Searching Agent (DSA)

To identify the one audio commercial the distances between pattern commercial matrix vectors and identifying commercial matrix vectors are computing by

$$d_i(YM_1, YM_2) = \sum_l |YM_1(i, l) - YM_2(i, l)| \tag{7}$$

for $i = 0, 1, \dots, K - 1$ and l changes from 1 to matrix row length. If this distance for one or more i is higher than establish value the recognition is negative. In this case DSA sends an alert message to the management agent. MA may proceed as follows:

- MA may instruct FEA to re-parameterization of the signal with different parameters
- MA may recognize unknown commercial as a new and add it to the database.

If recognition process is positive DSA sends the commercial database index to MA. The management agent creates the report of the identified advertisements (commercial name, time of broadcasting, etc.) which is sent to User.

3 Some Results

The presented system of the audio commercial recognition was implemented in C++ language and tested in PC environment. All the commercials were 8-bit signals sampled with frequency 11.2kHz. Audio fingerprints database was

Table 1. Influence of the filter parameters on the recognition effectiveness

Filter parameters	Recognition [%]
N=10, M=1	84
N=10, M=2	86
N=20, M=1	97
N=20, M=2	99
N=20, M=3	92
N=30, M=2	95

N - filter order, M - degree of nonlinearity.

created, which included 100 audio commercials (Polish TV commercials with a duration of 15 to 45 seconds).

Table 1 shows influence of the filter parameters (filter order N and nonlinearity degree M) on recognition effectiveness. Increasing values of the filter parameters above some values does not correct the recognition effectiveness, but can make it worse. It seems that filter order about $N = 20$ and degree nonlinearity $M = 2$ is the best solution. This finding was confirmed for another set of test tracks (the results differed by about 0.5 %).

In order to compare the proposed solution with existing audio fingerprint methods one selected to test the following audio signal parameterization algorithms:

- Robust Audio Hashing (RAH) [10]
- Normalized Spectral Subband Moments (NSSM) [12]
- Mel-Frequency Cepstrum Coefficients (MFCC) [11] [4]

Selecting the best result ($N = 20$, $M = 2$) one compared it with the results of other methods for parameterization audio tracks. The results are shown in table 2.

Table 2. Comparison of selected methods of parameterization

Method	Recognition [%]
RAH	98
NSSM	92
MFCC	95
Proposed method	99

There is no faultless method. Commercials were recorded in conditions different from their patterns, and this could hamper the identification. In addition, ads differ sometimes only one word which increased recognition errors.

The proposed multi-agent system can operate in real time. The process of signal parameterization of the average television advertising, which lasts 30 seconds, takes approximately 10 seconds for $N=20$ and $M=2$ (result for PC computer,

CPU AMD Athlon 2,4GHz). Searching a database consisting of 100 tracks takes approximately one second. So it is possible to keep track of advertisements and identify a single commercial in real time.

4 Conclusions

This paper proposed the autonomous commercial recognition system based on collected TV broadcasting signal. In the system multi-agents are involved in subsequent operations, such as advertising block detection, single commercial detection, ad features extraction, database searching. Presented system is efficient in more than 98% for not corrupted audio tracks. It is resistant on the poor recording conditions. Proper features extraction parameters selection is necessary for the high recognition effectiveness. Introducing multi-agent technology to the commercial recognition system proposed in this paper is expected to create high value-added ubiquitous services.

The proposed solution is fully scalable. Enlarge the database with additional advertising should not affect the degree of recognition. However, it may take longer time to search the database. In this case, you may consider removing some modules of the system to other computers connected to the network. This is a topic for further work on this system. The proposed solution has a modular architecture which allows exchange of an extraction unit for other parameters, giving the other features of the audio signal.

References

1. Kim, D.S.: Design and Implementation of Intelligent Agent System of Pattern Classification. *Korea fuzzy logic and intelligent society* 11(7), 598–602 (2001)
2. Kim, D.S., Kim, C.S., Rim, K.W.: Modeling and Design of Intelligent Agent System. *International Journal of Control, Automation and Systems* 1(2) (June 2003)
3. Hakamura, Y., Nagao, M.: Parallel Feature Extraction System with Multi Agents. In: *Proceedings of 11th IAPR Conference B: Pattern Recognition Methodology and Systems, Netherlands, vol. II* (1992)
4. Kay, S.M.: *Modern Spectral Estimation*. Prentice Hall, Englewood Cliffs (1988)
5. Rabiner, L.: *Fundamentals of Speech Recognition*. Prentice Hall PTR, Englewood Cliffs (1993)
6. Lienhart, R., Kuhmunch, C., Euelsberg, W.: On the detection and recognition of television commercials. In: *Proc. IEEE Conf. on Multimedia Computing and Systems, Ottawa, Canada, pp. 509–516* (June 1997)
7. Zabih, R., Miller, J., Mai, K.: A feature-based algorithm for detecting and classifying scene breaks. In: *ACM Conference on Multimedia, San Francisco, California* (November 1995)
8. Biernacki, P., Zarzycki, J.: Orthogonal Schur-Type Solution of the Nonlinear Noise-Cancelling Problem. In: *Proc. Int. Conf. On Signals and Electronic Systems (ICSES 2000), Ustron, pp. 337–342* (2000)

9. Lee, D.T.L., Morf, M., Friedlander, B.: Recursive Least-Squares Ladder Estimation Algorithms. *IEEE Trans. on CAS* 28, 467–481 (1981)
10. Haitsma, J., Kalker, T., Oostveen, J.: Robust audio hashing for content identification. In: *Proc. of the Content-Based Multimedia Indexing*, Firenze, Italy (September 2001)
11. Morgan, N., Boulard, H., Hermansky, H.: Automatic Speech Recognition: An Auditory Perspective. In: Greenberg, S., Ainsworth, W.A. (eds.) *Speech Processing in the Auditory System*, p. 315. Springer, Heidelberg (2004); ISBN 9780387005904
12. Paliwal, K.K.: Spectral subband centroid features for speech recognition. In: *Proc. IEEE ICASSP*, pp. 617–620 (1998)

A Hybrid Interestingness Heuristic Approach for Attribute-Oriented Mining

Maybin Muyebe¹, Keeley Crockett¹, and John Keane²

¹ School of Computing, Mathematics and Digital Technology, John Dalton building, Manchester M1 5GD

² School of Computer Science, The University of Manchester, M13 9PL
{m.muyebe, k.crockett}@mmu.ac.uk,
john.keane@cs.manchester.ac.uk

Abstract. A hybrid interestingness heuristic algorithm, *clusterAOI*, is presented that generates a more interesting generalized final table than traditional attribute-oriented induction (AOI). AOI uses a global static threshold to generalize attributes irrespective of attribute features, consequently leading to overgeneralisation. In contrast, *clusterAOI* uses attribute features such as concept hierarchies and distinct domain attribute values to dynamically recalculate new attribute thresholds for each of the less significant attributes. *ClusterAOI* then applies new heuristic functions and the Kullback-leibler (K-L) measure to evaluate interestingness for each attribute and then for all attributes by a harmonic aggregation in each generalisation iteration. The dynamic threshold adjustment, aggregation and evaluation of interestingness within each generalization iteration ultimately generates a higher quality final table than traditional AOI. Results from real-world cancer and population datasets show both significantly increased interestingness and better performance compared with AOI.

Keywords: Induction, heuristic function, threshold, interestingness, generalization, cluster.

1 Introduction

A goal of data mining is to find interesting patterns [10] from large data in a scalable fashion that are both concise and interpretable by humans. Objectively, patterns are interesting if they satisfy user-supplied thresholds and the choice of such thresholds can vary considerably between users. The traditional attribute-oriented induction (AOI) algorithm uses global thresholds to limit the number of distinct values in the input table. These algorithms generalize attributes by replacing attribute values with concept hierarchy data (domain knowledge) [1][4] and then merging tuples to eliminate duplicate tuples as long as distinct attribute values are greater than a given threshold. The AOI algorithm was introduced by Han and colleagues [1][8]. Most attribute-oriented induction algorithms [5][6][7] use thresholds to determine distinct attribute values and evaluate pattern interestingness but do not use the inherent attribute features like the number of distinct values of certain clusters of related attribute values, concept hierarchy probability distributions of node values etc, which often

vary between attributes. In short, these algorithms are essentially one-size fits all. For example, attribute “Gender” (“M”, “F”) has a far lower distribution of values than say attribute “City” with values (“Aberdeen”, “Accra”, “Manchester”, “Toronto”). Threshold driven algorithms only stop when thresholds are satisfied, not necessarily when interesting patterns are found [5][11]. In the context of a user’s mining intention-data, thresholds, domain knowledge [4] etc, objective interestingness can still be improved. In AOI, the aim is to generate output with mostly interior concepts, thus avoiding excessive abstraction to value “ANY” (or overgeneralization) [12]. An interior concept is any concept hierarchy node value excluding all leaf nodes and the root. Further, [3] proposed a simple heuristic to define interestingness of a value based on its distance from the root node. Clearly, then, leaf concepts are furthest from the root and should be regarded as interesting. Again, depending on tree structure, distances may vary between node values and therefore may not present reliable and interesting patterns. The goal then is to generate a final table that ensures that there are more interior concepts than would be generated by traditional AOI. A few AOI algorithms address interestingness: external interestingness evaluates concept hierarchies before the mining process [4] (pre-AOI approach); others evaluate interestingness on the final table [1][10] (post- AOI approach).

This paper extends AOI by improving final table interestingness through a hybridized approach i.e. pre-, post- and internal clusterAOI methods. In pre- clusterAOI, attribute thresholds (or local thresholds) are recalculated. In internal clusterAOI, an attribute’s significance is found by calculating probabilistic semantic similarity [2] between clusters of attribute values. By this, clusterAOI performs fewer iterations and consequently generates fewer abstracted values and better runtime performance than traditional AOI. ClusterAOI also checks an attribute’s interestingness during the generalization process by a harmonic aggregation of certain heuristics about each attribute (see section 4). In post- clusterAOI, the final output table’s interestingness using both Kullback-Leibler [10] (divergence or “distance” between two probability distributions) and the global harmonic mean is calculated.

The paper is organized as follows: section 1.1 gives a motivating example, section 2 presents the problem definition; the pre-clusterAOI approach is discussed in section 3; section 4 considers internal-and post clusterAOI; section 5 presents experimentation and analysis; and finally conclusions are given in section 6.

1.1 Motivating Example

In this section, we demonstrate how clusterAOI handles overgeneralization better than traditional AOI irrespective of any dataset used, and so helps to improve interestingness of the final patterns. This is further shown in section 5 where two real datasets are used.

Table 1 is a football dataset with seven tuples and concept hierarchies for each of the attributes ‘diameter’ and ‘colour’ (figure 1). Let us set a global threshold, $G.Thr=2$. Analysing attributes in table 1 and figure 1, attribute “Diameter” has 7 distinct values and 7 concept hierarchy node values while “Colour” has 6 distinct values and 16 concept hierarchy node values. We ran traditional AOI and clusterAOI and compared the final table outputs (table 2). ClusterAOI recalculates local thresholds, $L.Thr$ (see table II). Attribute “Diameter” has a new threshold value $L.Thr=3$, while “Colour” still has global threshold $L.Thr=G.Thr=2$. This is a multi-threshold approach explained as follows:

Table 1. Ball data

Diameter	Colour
2	Red
2	Blue
34	Yellow
25	Green
28	Orange
8	Violet
16	Red

Table 2. Ball data, original threshold=2

Algorithm	Diameter	Colour	Count
AOI	Any	Non-dark	5
	Any	Dark	2
G.Thr=2	L.Thr=2	L.Thr=2	4
	Medium	Non-dark	4
clusterAOI	Medium	Bluish	2
	Small	Non-dark	1
G.Thr=2	L.Thr=3	L.Thr=2	

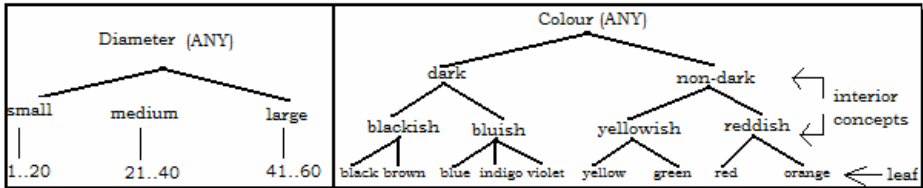


Fig. 1. Concept hierarchies for Diameter and Colour

We can see that 50% (5+2=7 out of 14) of attribute values for AOI are overgeneralised i.e. seven “ANY” values from “Diameter” are generated. In contrast, clusterAOI has 0% overgeneralisation i.e. zero “ANY” values exist. Based on the structures and attribute value distributions in table 1 and figure 1, intuitively “Diameter” is less significant than “Colour” in terms of total information contained. ClusterAOI uses this principle and then recalculates thresholds. In a practical sense, “Colour” is more significant than “Diameter” and will have more generalizations than “Diameter”. Further, we set a global threshold of 1 using table 1. AOI overgeneralises 100% to one tuple {“ANY”, “ANY”, 7} while clusterAOI generalized to two tuples {“ANY”, “non-Dark”, 5} and {“ANY”, “Dark”, 2} giving only 50% overgeneralisation. This impacts enormously on pattern interestingness and their intelligent interpretation. In some cases where both AOI and clusterAOI have 0% over generalisations, clusterAOI is still significantly superior to AOI (see section 4.2).

The motivation for the work in this paper is based on the following observations:

- (1) In all the cited work, global thresholds are not adjusted dynamically based on an attribute’s features
- (2) Current AOI algorithms, and many others, do not evaluate attribute or overall interestingness during the mining process [10]
- (3) From our analysis of AOI, we need to consider both internal and external interestingness in a hybridized fashion. For external interestingness, we evaluate new thresholds based on attribute features, and for post-AOI we evaluate the final table using the K-L measure and an aggregate harmonic mean function. The latter measures an attribute’s internal interestingness in each local generalization iteration before calculating the global harmonic mean of all attributes.

2 Problem Definition

The interestingness problem in clusterAOI is as follows: Let a relation R be defined on dataset $D \subseteq R$ with n tuples, attributes A_i and attribute hierarchy pairs exist for m attributes i.e. $\{(A_1, H_1), (A_2, H_2), \dots, (A_m, H_m)\}$ A_{m+1} is a vote count attribute of the number of tuples in R and t a global threshold. Then $\sum |A_{m+1}, \emptyset| = n$, with domain values $Dom(A_{m+1}, \emptyset) \in Z^+$.

The first problem is to find attribute significance (entropy values) from concept hierarchies and distinct values for each attribute. We define a generalization space $B_i = A_i \cup H_i$ for each attribute. We use a function $\nabla(A_i)_i$ on the generalization space and then use the entropy values in a linear function to find thresholds t_i per attribute. Then with description language $L = (B_i, f)$, a level by level mapping function $f : B_i \rightarrow Dom(H_i)$ is a generalization function “nearest parent” and a partial order on B_i i.e. (\prec, B_i) . Generalization stops when an attribute has reached a local interestingness value. The same holds for the global case (see section 4.2).

The problem of mining interesting generalized patterns is defined as follows, given dataset D , and final table $T \subset B_i$ hierarchies H_i , attributes A_i and space B :

- (1) Find attribute significance using $\nabla(A_i)$ and generate a new thresholds t_i
- (2) Find local attribute interestingness and aggregate values using function I^i_L .
- (3) Generalise attribute values using the function $f(B_i)$ subject to t_i .
- (4) Find the global interestingness of T by aggregation of local I^i_L values using function I^T_g .

The problem is a 4-tuple $(\nabla, I^i_L, f, I^T_g)$ (1) to (4). We define and describe each function in the following sections: 3.1 and 3.2 presents (1), 4.1 presents (2), and 4.2 deals with (3) and (4).

3 Pre-clusterAOI

3.1 Attribute Feature Significance

We calculate each attribute’s significance using attribute features (distinct values in D and concept hierarchy node values in H_i) or based on an attribute’s information content [2]. Entropy e_i of attribute A_i where each c_i is a concept hierarchy node value is used to find attribute significance by a ranking function ∇ :

$$e_i = \nabla(A_i) = -\left(\sum_i^m \log p(c_i) + \log \frac{|D|}{\text{dist}(D, A_i)}\right) \tag{1}$$

where $\text{dist}(D, A_i)$ is the number of distinct values of an attribute, taking the absolute value of equation 1. Note that $c_i \in H_i$ uses no leaf concepts in the first part of equation 1. From 1, attribute significance values are calculated as follows:

$$e_{\text{diameter}} = \nabla(A_{\text{diameter}}) = 4.75532$$

Similarly, $e_{\text{colour}} = 9.77760$, meaning that ‘‘Colour’’ is more significant than ‘‘Diameter’’. From this, new thresholds can be calculated.

3.2 Multi-threshold Generation

After finding attribute significance, we can generate different local thresholds ($L.Thr_i$) using entropy values e_i for each attribute A_i by a linear function in e_i :

$$L.Thr_i = \frac{(G.Thr * E) + |E - e_i|}{E} = G.Thr + \frac{|E - e_i|}{E} \tag{2}$$

where $E = \text{MAX}(e_1, e_2, \dots, e_m)$. Equation 2 simply states that $L.Thr_i \geq G.Thr$ for every attribute meaning that every attribute will be generalized using, in the worst case, the global threshold. Hence, choosing the maximum entropy value is justified. Using table 1 and a global threshold $G.Thr = 2$, we obtain entropy values as before, Diameter $e_1 = 4.75532$ and Colour $e_2 = 9.77760$. Local thresholds $L.Thr_i$ are calculated as follows:

$$L.Thr_{\text{colour}} = \frac{(2 * 9.77760 + |9.77760 - 9.77760|)}{9.77760} = G.Thr = 2$$

$$L.Thr_{\text{diameter}} = \frac{(2 * 9.77760 + |9.77760 - 4.75532|)}{9.77760} = 2.51366 > G.Thr$$

where $\text{Max}(4.7553, 9.77760) = 9.77760$. Rounding off the values, attribute ‘‘Colour’’ will be generalized using $L.Thr_{\text{colour}} = 2$ as $|E - e_i| = 0$ while ‘‘Diameter’’ $L.Thr_{\text{diameter}} = 3$. This approach illustrates multi-threshold generation of local attribute thresholds, which affects each attribute’s interestingness.

4 Internal clusterAOI Interestingness

4.1 Attribute Interestingness

We know that internal tree nodes are more interesting than both leaf nodes and root node ‘‘ANY’’ [3][4], hence this clearly represents a bounded interestingness problem.

We do not attempt to establish its theoretical underpinnings in this paper. As generally traditional AOI [1] uses a distance-driven generalization, interestingness should naturally be a distance function i.e. how far a value is from its root $d(c, "ANY")$ or leaf level [3]. The initial rationale is that higher level concepts are more interesting than primitive ones [4]. Contrary to this assertion, root node "ANY" is not interesting so we deal with this notion using the probabilistic equation 4. Using equation (3) from [3], the interestingness function is a mapping $I_c : [0, 1] \rightarrow [0, 1]$.

$$I_c = 1 - \frac{d(c, "ANY")}{depth("ANY")}, c \neq "ANY" \tag{3}$$

For example, a leaf concept $x_i \in A_i$ for attribute A_i has interestingness value $I_{x_i} = 1 - \frac{d(x_i, "ANY")}{depth("ANY")} = 0$. Similarly, $I_{c=ANY} = 1 - 0 = 1$. This demonstrates how unreliable distance measures are in determining interestingness of hierarchical data. i.e. value "ANY" is least interesting but has the maximum value of 1 by the mapping function. We conclude that interestingness is non-linearly distributed between leaf and root nodes. This is calculated using equation 4 with input from equation 3.

$$H(I_c) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{(2\sigma^2)}} \tag{4}$$

where $0 \leq I_c \leq 1$ with mean and variance values (μ, σ) . Equation 4 then becomes our new interestingness function, $\forall c \in B_i$. Using table 1, with mean and variance $\mu = 0.524, \sigma = 0.069$ for "Colour" and $\mu = 0.625, \sigma = 0.0625$ for "Diameter", we can observe that equation 4 gives $H(I_{red}) = H(0) = 1.48 \times 10^{-12}$ while $H(I_{ANY}) = H(1) = 0.022 \times 10^{-12}$. For value "reddish" with interestingness value $I_{reddish} = 0.333$, we obtain $H(I_{reddish}) = H(0.333) = 0.124$. Notice that $H(I_{reddish}) \geq H(I_{red}) \geq H(I_{ANY})$ thus confirming that root and leaf node values are less interesting than interior concept values. ClusterOAI therefore aims to generalize values to interior concepts. The generalization process follows.

4.2 Generalisation Process

We follow a similar AOI clustering approach to that given in [3] i.e. by clustering attribute values using the nearest parent of each value. In table 1, the value "small" is the nearest parent for the cluster {2, 7, 8, 16} so these are generalized to their parent "small" as there are 4 distinct values i.e. $4 > L.Thr_{diameter} = 3$. Adopting this approach ensures that we only generalize minimally, cluster by cluster. ClusterAOI depends largely on this step to produce better patterns. Let C_{ij} be cluster j for a group of

attribute values for attribute i , $1 \leq j \leq n$, for n tuples in a table and $C = \{\bigcup_{j=1}^k C_{ij}\}$ is a collection of all clusters for an attribute and becomes smaller as further generalization occurs. In addition, attribute properties become similar too, e.g. equal distances from the root node and same parent nodes. Using heuristic functions defined in [3], namely, to find similarity within clusters (IntraC), tightness in clusters (InterC) and total local interestingness (I_L^i), we can find an attribute's global interestingness value (equation 5).

$$I_g^i(X) = \frac{n}{\sum_{i=1}^n 1/x_i} \tag{5}$$

where $X = (x_1, x_2, x_3)$ are Intra, Inter and total cluster interestingness values respectively. After computing I_g^i , we store the global average interestingness value I_g^D for all table attributes (equation 6). To do further generalization on any attribute, the algorithm compares the latest and previous I_g^i to determine the next step. If the latest value is larger (i.e. highest interestingness so far), generalization continues, otherwise the attribute will not be generalized further. Comparing values this way avoids overgeneralization. Similarly, global interestingness values (equation 6) are compared in the same way. The cluster quality of any table T is a summation of all local I_L^i values (equation 7)

$$I_g^D = \frac{\sum_{i=1}^m I_g^i}{m} \tag{6}$$

$$I_g^T = \sum_{i=1}^m I_L^i \tag{7}$$

The clusterAOI algorithm is shown below:

ClusterAOI algorithm:

Input: A dataset D , attributes A_i , $i=1, \dots, m$; Concept hierarchies Chi , threshold t

Output: A compact table T , Kullback measure $K-L$, Cluster measure I_g^T

BEGIN

Step 1, 2. Select attributes A_i , Calculate attribute significance, new thresholds
 $iter \leftarrow 1$

Step 3. Process attributes

stillInteresting=true

convergenceValues [A_i] = false

```

While (stillInteresting) // check Global interestingness
  For each Ai in V
    attrConv= convergenceValues (Ai)
    if ( not(attrConv))
      D←processAttribute(D, Chi, Ai)
      D←mergeData(D)
      attrHeur[i]←computeHeuristics(interC,intraC, ILi )
      hashStore=IterationObject(attrHeur[i], iter)
      harm_total←0
      Igi ←0
  For each Ai //check attribute convergence
    if(iter>0)
      attrConv←isAttribConv(iter, i, hashStore)
      if(attrConv= true)// Check Local interestingness
        convergenceValues(i)←attrConv
        Igi ← computeGlobal_I(hashStore (iter, Ai))
        harm_total← harm_total +Igi
  IgD =harm_total/m //eqn (6)
  setHarmValues(IgD )
  stillInteresting←isInteresting(getHarmValues())
  iter←iter+1
Step 4. Generate final table
  D←merge and Copy(D)
  T←topKRules(D, newThreshold)
  K-L←KullbackMeasure(T)
  IgT←TableClusterQuality(T) //eqn (7)
  Output(T, K-L, IgT)
END

```

In step 3 of the algorithm, we set global interestingness, process each attribute, merge resulting values in the table and store heuristic values i.e. IntraC, InterC, local interestingness I_L^i values. ProcessAttribute() in each iteration generalizes attributes by finding the common and nearest parent of a cluster of values and evaluates them against a threshold using makeParentClusters(). The interestingness heuristics including equation 5 are computed and stored by invoking computeHeuristics(). We then check each attribute's local convergence, and later the global convergence of the algorithm using the stored heuristics. The global interestingness value is checked by the global harmonic mean of all local interestingness values (equation 6). Step 4. merges the table rows and copies them to a new smaller table. Table 3 shows initial clusters generated. The set of global harmonic means are stored for the whole generalization process as sets of vector values $[h_1, h_2, \dots] = [0.0, 0.89, 0.91, 0.97, \dots]$ with table 3 showing the first iteration result as 0.89. Local interestingness "Diameter" are calculated as $[0.0, 0.58, 1.03, 0.97]$ and for "Colour" $[0.0, 0.1.2, \dots]$.

Table 3. Clusters for “Diameter” and “Colour” - Iteration 1

Diameter	<i>IntraC</i>	<i>InterC</i>	I^i_L	I^i_g (eqn. 5)
[2, 7, 8, 16]	0.25	0.0	0.0	
[34, 25, 28]	0.33	0.0	0.0	
TOTAL	0.58	0.0	0.0	0.58
Colour				
[red, orange, red]	0.5	1.0	0.0	
[blue, violet]	0.5	0.0	0.0	
[yellow, green]	0.5	0.0	0.0	
TOTAL	1.5	1.0	0.0	1.2
<i>Average H. Mean (eqn. 6)</i>				1.78/2=0.890

As “Diameter” has a new lower value of 0.97 compared to 1.03, generalization stops. The same criteria is used for stopping generalisation in the global case.

4.3 Post-clusterAOI Interestingness

Table 4 compares the final output from both algorithms. Clearly, clusterAOI is superior to AOI: global interestingness (I^T_g , 5th column) is 3 times better and K-L is 1.6 times better than AOI. Further, we have discussed overgeneralization comparisons in section 1.1.

Table 4. ClusterAOI and AOI final table comparisons

	<i>Diameter</i>	<i>Colour</i>	<i>Count</i>	I^T_g	<i>K-L</i>
ClusterAOI	Medium	Non-Dark	4	4.06	1.38
	Small	Bluish	2		
	Small	Non-Dark	1		
AOI	<i>Diameter</i>	<i>Colour</i>	<i>Count</i>	1.34	0.86
	ANY	Non-Dark	5		
	ANY	Dark	2		

5 Experimental Analysis

We have evaluated both AOI and clusterAOI on two real datasets of different sizes: a real census-income 50K population (3 attributes-age, education, number worked) and a 700 record cancer dataset (4 attributes- cell size, bare & normal nucleus, mitosis) from the UCI repository with respective concept hierarchies. Experiments were run 5 times to obtain average results. Table 5 gives the results for different threshold values 1 to 5, cluster quality and the K-L measures. For ClusterAOI, mean and variance for each attribute were calculated and their significance evaluated as $sig(age) = 18.873$, $sig(educ) = 9.986$ and $sig(numWorked) = -1.632$. Global thresholds were recalculated as before (see table 5, last column).

Table 5. Interestingness measure I^T_g , Kullback-Leibler (K-L) on 50K population data set

G.Thr	AOI		ClusterAOI		L.Thr
	I^T_g	K-L	I^T_g	K-L	
1	0.000	0.000	0.076	1.55	1,2,3
2	0.001	0.863	0.230	1.86	2,3,4
3	0.0018	1.461	0.315	2.14	3,4,5
4	0.0018	1.461	0.396	2.38	4,5,6
5	0.0030	2.260	0.360	2.35	5,6,7

Comparing I^T_g values or global interestingness (cluster quality) in table 5 (also figure 2(b)), ClusterAOI is 36 times better than AOI. Similarly, the K-L value for clusterAOI is 1.57 times better than traditional AOI (see figure 2(a)). The results indicate that clusterAOI generalizes better and produces a wider variety of patterns than those obtainable from AOI.

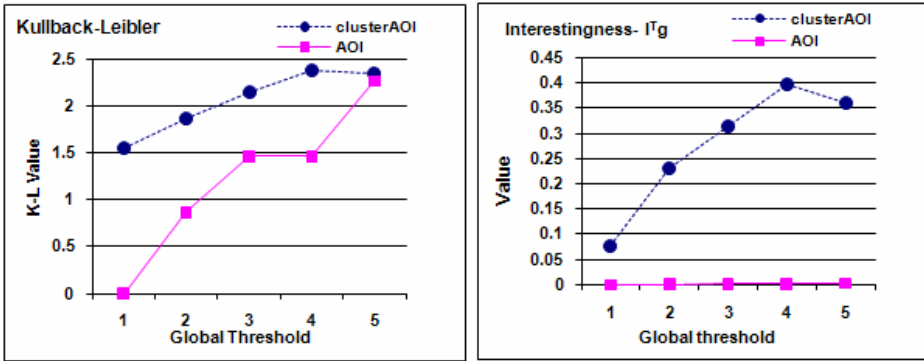


Fig. 2. (a) K-L on 50K population data (b) Global Interestingness on 50K population data

For example, for cancer data, AOI had 50% overgeneralization while clusterAOI had 0%. Comparing one tuple (i.e. a rule) of final cancer data output tables between the two algorithms, the rule showed 70% of data had average cancer cell sizes and average bare and normal nucleus sizes irrespective of mitoses (i.e. mitoses were overgeneralised). In contrast, clusterAOI showed 67% of the data had average cancer cell sizes, average bare and normal nucleus sizes, and average mitoses. These differences indicate that clusterAOI generates both more intelligent and informative patterns than does AOI. For intelligent decision making, clusterAOI has greater applicability in the real world. Figure 2(b) confirms these notable differences. Similarly, clusterAOI is 1.5 times faster than traditional AOI on average on the 50K population dataset. This improvement seems to be a consequence of clusterAOI’s new hybrid 3-fold strategy (pre-, post- and internal-) and usage of new thresholds.

6 Conclusion

We have introduced an interestingness heuristic algorithm, clusterAOI that has increased interestingness and better runtime performance than traditional AOI. ClusterAOI is not entirely a threshold-driven algorithm but rather uses pre-, internal and post- strategies to solve the interestingness problem in traditional AOI. ClusterAOI uses heuristics to find attribute significance and then local and global interestingness. Results show that clusterAOI performs on average 1.57 times higher on Kullback-Leibler and is 1.5 times faster in terms of run-time than traditional AOI.

The interestingness evaluation presented here should be applicable to various generalization algorithms that use concept hierarchies [9]. Clearly, we can also use optimization algorithms like Genetic algorithms to handle local and global optimum interestingness or statistical approaches like maximum likelihood estimate.

References

1. Han, J., Cercone, N., Cai, Y.: Attribute-Oriented Induction in Relational Databases. In: Pi-atetsky-Shapiro, G., Frawley, W.J. (eds.) *Knowledge Discovery in Databases*, pp. 213–228 (1991)
2. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy (1). In: *Proc.14th Joint Conference on Artificial intelligence*, San Francisco, vol. 1, pp. 448–453 (1995)
3. Muyebe, M., Sulaiman Khan, M., Gong, Z.: On Clustering Attribute-oriented Induction. In: *Research and Development in Intelligent Systems XXIII 2006*, pp. 403–407 (2006)
4. Fudger, D.R., Hamilton, J.: A Heuristic for Evaluating Databases for Knowledge Discovery with DBLEARN. *Rough Sets and Knowledge Discovery*, pp. 29–39 (1993)
5. Chen, Y.L., Wu, Y.Y., Chang, R.I.: From data to global generalized knowledge. *Journal of Knowledge and Information Systems* (2010) (in Press)
6. Yen-Liang, C., Ching-Cheng, S.: Mining generalized knowledge from ordered data through attribute-oriented induction techniques. *European Journal of Operational Research* 166(1), 221–245 (2005)
7. Hsu, C.-C.: Extending attribute-oriented induction algorithm for major values and numeric values. *Expert Systems with Applications* 27(2), 187–202 (2004)
8. Han, J., Fu, Y.: Exploration of the power of attribute-oriented induction in data mining. In: *Advances in Knowledge Discovery and Data Mining* (1996)
9. Huang, Y.-F., Wu, C.-M.: Mining Generalized Association Rules Using Pruning Techniques. In: *ICDM 2002*, pp. 227–234 (2002)
10. Hilderman, R.J., Hamilton, H.J.: *Knowledge Discovery and Measures of Interest*. Kluwer Academic Publishers, Dordrecht (2001)
11. Silberschatz, A., Tuzhilin, A.: What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering* 8(6), 970–974 (1996)
12. Kim, J., Lee, G., Seo, J.-T., Park, E.-K., Park, C.-s., Kim, D.-K.: An Alert Reasoning Method for Intrusion Detection System Using Attribute Oriented Induction. In: Kim, C. (ed.) *ICOIN 2005*. LNCS, vol. 3391, pp. 72–79. Springer, Heidelberg (2005)

The Network Balance Realized by Routing Organization System

Henryk Piech¹ and Adam Czubak²

¹ Institute of Computer and Information Sciences
Czestochowa University of Technology, Poland

`h.piech@adm.pcz.czest.pl`

² Institute of Mathematics and Computer Science
Opole University, Poland

`adam.czubak@math.uni.opole.pl`

Abstract. In the presented paper, we propose to exploit routing organization for the purpose of managing network resources. According to our assumptions, we have the same quantity of materials, objects, tokens, tools etc. at our disposal in every node of the network. During network operation we must distribute resources between these nodes. It should be carried out as instantaneously and as economically as possible. Multi-Agent Systems are also used to deal with this kind of tasks and the centralised algorithms presented in this paper are to be used to measure the efficiency of the distributed MAS solution. From the logistical point of view, we have a sequence of stages with different states of tokens (objects). The criterion of optimal redeployment among nodes is, as usual, minimal cost and, indirectly, time as well. The levels of resources in nodes change as the node role changes from the supplier to the receiver and inversely. In practice, there are many applications of such tasks in logistics, military regrouping, elaboration of evacuation strategy, warehouse management etc. The distances between nodes can be fixed or may vary. This varied approach does not complicate our problem essentially, however we should remember, that this parameter influences the total redeployment cost.

Keywords: routing strategy, network balance, network optimization.

1 Introduction

The network balance problem can be realized with the help of a routing strategy. Routing is a fundamental function of a network. The authors of [1,2] show, that sending tokens with different routing strategies can ensure that one of the solutions will be optimal according the chosen criterion (for multicriteria optimization see [3]). At each node and for any destination the algorithms require as little as constant time and space to determine the next node to which the objects (tokens, messages) are to be sent [4]. It is desirable to route each of the objects along the shortest path from its source to destination. To improve

the performance of the network many fault-tolerant routing algorithms have been proposed. For example, Escudero et al. presented an algorithm by finding a node that can be used as a relay node [5], Horton et al. presented an algorithm for some special kinds of networks which have small diameters [6], and Guan [7] presented an algorithm by checking, whether the objects are on the shortest path or not. Adaptation of classical Bellman-Ford approaches have been proposed as well [8]. All these algorithms assume that the element is known at the time, the algorithm decides by which edge the message is to be sent. In practical considerations however, it may happen that the objects are directed only after global analysis and no distributed approach is required. In this paper we present an optimal routing regarding all sent parts of objects. Thus, we assume that the shortest distances between the nodes chosen by algorithm are known. Our strategy can be used in both: the projecting and exploiting the phase of network balancing. When the distances among the nodes change, we enrich the algorithms with dynamic parameter definitions but it does not have an influence on the general strategy.

2 Definition of Network Balance and Its Applications

Generally, we can say that network balance is based on redeploying objects (state attributes, tokens) by way of changing the predefined needs in the nodes (places). It should be realized with minimal total costs in all the following stages. Let us consider 4-tuple (P, T, F, c) , where P - is a set of places, $T = RT \cup DT$ - set of tokens sets for all m stages, $RT = \bigcup t^{(k)}(i)$ - set of real tokens values, $DT = \bigcup dt^{(k)}(i)$ - set of predefined tokens level, $k = 1, 2, \dots, m$ - state number, $i = 1, 2, \dots, n$ - node number. Generally, we assume that real token levels should be not less than the needs: $t^{(k)}(i) \geq dt^{(k)}(i)$. It is possible to fulfil when $\sum_{i=1}^n t^{(k)}(i) \geq \sum_{i=1}^n dt^{(k)}(i)$. In contrast, we proportionally decrease the assumed values of the needs: $dt'^{(k)}(i) = dt^{(k)}(i) * (\sum_{i=1}^n t^{(k)}(i) / \sum_{i=1}^n dt^{(k)}(i))$. The flow levels $f(i, j)$ refer to $P \times P$ edges (connections), and are validated when the total cost of redeployment is adequate to minimal $c = g(f(i, j), d(i, j)) \rightarrow \min$ where the cost function $g(*)$ is the distance $d(i, j)$ between nodes i and j . Hence, the cost is a criterion for optimal network balance. The routing system organization can be described by exploiting tables or more effectively with floating tables:

0	$f^{(k)}(1, 2)$	(...)	$f^{(k)}(1, n)$
$f^{(k)}(2, 1)$	0	(...)	$f^{(k)}(2, n)$
(...)	(...)	(...)	(...)
$f^{(k)}(n, 1)$	$f^{(k)}(n, 2)$	(...)	0

where $k = 1, 2, \dots, m$. The set of nodes P contains information about nodes locations (coordinates) or distances among nodes (stable or variable values in particular stages). There are different types of task definitions, e.g.:

1. Assign tables of $dt^{(k)}(i)$ and $d(i, j) = const$ find $f^{(k)}(i, j)$ for $c \rightarrow min$ (season redeploying products or resources in warehouses to different places),
2. Assign tables of $dt^{(k)}(i)$ and $d(i, j) = var$ (in sequenced stages) find $f^{(k)}(i, j)$ for $c \rightarrow min$ (regrouping soldiers and arms during war operations),
3. Assign tables of $dt^{(k)}(i)$, $c = dc(given level)$, $d(i, j) = const$ find $f^{(k)}(i, j)$ (possibility of obtaining greater set of solutions and include additional, not necessarily dislocating, criteria),
4. Assign tables of $dt^{(k)}(i)$, $c = dc(given level)$, $d(i, j) = var$ find $f^{(k)}(i, j)$,
5. Assign tables of $dt^{(k)}(i)$, $c = dc(given level)$, $f(i, j) = const$ find $d^{(k)}(i, j)$ (seeking optimal location for evacuation center based on possibility of influence on total cost by distance (among nodes) regulation),
6. Assign tables of $dt^{(k)}(i)$, $de^{(k)}(i)$ (part of "consumed" tokens in every stage), $d(i, j) = const$ find $f^{(k)}(i, j)$ for $c \rightarrow min$ (storeroom management with regard to consumption and supplies).

3 The Idea of Network Balance Presentation

Let us assume, that every node of a network is associated with a state attribute (or token number in Petri net terminology) $t(i)$. Additionally, we assume that every edge has an associated weight $w(i, j)$. The value of the flow level according to PT-net assumptions can be determined in different ways e.g.:

1. $f(i, j) = \min\{t(i), w(i, j)\}$
2. $f(i, j) = w(i, j)$ if $f(i, j) \geq w(i, j)$ and $f(i, j) = 0$ if $f(i, j) < w(i, j)$
3. $f(i, j) = dw(i, j)$ where $dw(i, j) < \min\{t(i), w(i, j)\}$ - designated (predefined) value of flow, $f(i, j) = -f(j, i)$ - flow level.

In our assumption, the weights of edges $w(i, j)$ will not be regarded because we are in the projecting phase of creating a network structure and its parameters as well as a routing organization system. In practice, after analysing all of the states, we can choose design parameters: $w(i, j) = \max_{k=1,2,\dots,m} \{f^{(k)}(i, j)\}$. The reasons for the flow between nodes are the requirements concerning the objects, products, arms, information etc. in a particular moment for a particular node. To simplify the problem description, we assume that the requirements refer to the levels of state attributes in the next stage $t^{(1)}(i)$, $i = 1, 2, \dots, n$. The requirements will be satisfied when the real value of state attributes is greater than or equal to the predefined levels of corresponding requirements: $t^{(1)}(i) \geq dt^{(1)}(i)$, $i = 1, 2, \dots, n$. We can name the set of state attributes a *balance spectrum*. While the $m + 1$ stages will be conducted, the descriptions of the spectra will be following:

$t(1)$	$t(2)$	(\dots)	$t(n)$
$t^{(1)}(1)$	$t^{(1)}(2)$	(\dots)	$t^{(1)}(n)$
(\dots)	(\dots)	(\dots)	(\dots)
$t^{(m)}(1)$	$t^{(m)}(2)$	(\dots)	$t^{(m)}(n)$

When we consider only two stages, the main goal of network balancing is moving transition tokens to the next stage with minimal costs. Thus, achieving the balance relies on the most effective redeployment of objects. Obviously, it must be realized for all nodes. For all $m + 1$ stages we have to regard the complete range of the state attributes. The cost between the two nodes depends on the distance and flow value $c(i, j) = d(i, j)|f(i, j)|$. The general cost for all the stages and routing connections is therefore equal to:

$$c = \sum_{k=1}^m \sum_{i=1}^{n-2} \sum_{j=i+1}^n d(i, j) * |f^{(k)}(i, j)|$$

We present four examples of balancing in the case of two and three stages. We start with the spectra for two stages:

$$t(1) = 6, t(2) = 2, t(3) = 5, t(4) = 15, \\ dt^{(1)}(1) = 1, dt^{(1)}(2) = 5, dt^{(1)}(3) = 10, dt^{(1)}(4) = 3$$

for three stages:

$$t(1) = 6, t(2) = 2, t(3) = 5, t(4) = 15, \\ dt^{(1)}(1) = 1, dt^{(1)}(2) = 5, dt^{(1)}(3) = 10, dt^{(1)}(4) = 3, \\ dt^{(2)}(1) = 0, dt^{(2)}(2) = 7, dt^{(2)}(3) = 12, dt^{(2)}(4) = 1,$$

and distances:

$$d(1, 2) = 2, 2; d(1, 3) = 4, 2; d(1, 4) = 11; \\ d(2, 3) = 2, 2; d(2, 4) = 10; d(3, 4) = 8, 2.$$

To compare the situation for two and three stages, we search for the minimal cost of flows to achieve balances and obtain for two stages:

$$f(1, 2) = 3; f(1, 3) = 2; f(3, 4) = -3, \\ t^{(1)}(1) = 1, t^{(1)}(2) = 5, t^{(1)}(3) = 10, t^{(1)}(4) = 12,$$

and adequate minimal cost:

$$c_{II} = \underline{2, 2 * 3 + 4, 2 * 2 + 8, 2 * 3} = 39, 6,$$

for three stages:

$$f(1, 2) = 3; f(3, 4) = -5, \\ t^{(1)}(1) = 3, t^{(1)}(2) = 5, t^{(1)}(3) = 10, t^{(1)}(4) = 12, \\ f(1, 2) = 2; f(3, 4) = -2, \\ t^{(2)}(1) = 1, t^{(2)}(2) = 7, t^{(2)}(3) = 12, t^{(2)}(4) = 10,$$

and adequate minimal cost:

$$c_{III} = \underline{2, 2 * 3 + 8, 2 * 5} + 2, 2 * 2 + 8, 2 * 2 = 68, 4.$$

The presented example shows that when adding stages, we have to use a different strategy regarding routing connections (it corresponds to the underlined parts of costs: $c_{II}^{(1)} < c_{III}^{(1)}$, under the same preliminary defined condition according to dt^{+1}). We describe our optimization task as follows:

$$c = \sum_{k=1}^m \sum_{i=1}^{n-2} \sum_{j=i+1}^n d(i, j) * |f^{(k)}(i, j)| \rightarrow \min$$

with following constrains:

$$t^{(k)}(i) \geq dt^{(k)}(i); \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, m$$

An additional limitation is required due to the initial token values:

$$\sum_{i=1}^n t^{(0)}(i) \geq \sum_{i=1}^n t^{(k)}(i); \quad k = 1, 2, \dots, m.$$

In real-word applications tokens (state attributes levels) can be exploited or consumed. Therefore we modify the last expression:

$$\sum_{i=1}^n (t^{(k-1)}(i) - e^{(k-1)}(i)) \geq \sum_{i=1}^n (t^{(k)}(i) - e^{(k)}(i)); \quad k = 1, 2, \dots, m,$$

where $e^{(k)}(i)$ - value of consumption of attribute i in stage k and

$$\sum_{r=1}^k \sum_{i=1}^n (t^{(r-1)}(i) - e^{(r-1)}(i)) \geq \sum_{r=1}^k \sum_{i=1}^n (t^{(r)}(i) - e^{(r)}(i)); \quad k = 1, 2, \dots, m.$$

4 Data Preparation for Optimization Procedures

We wish to propose the following data structures in Table 1. A data flowing example is constructed with the help of arrays. In the presented variant, the distances among the nodes are constant [see Table 2]. When stages are connected with consumption, then operation: $t^{(k)}(i) = t^{(k)}(i) - e^{(k)}(i)$ is realized between arrays (stages). The values of $f^{(k)}(i, j)$ are obviously the results of optimization.

5 Optimization Process in Network Balancing

To prepare for optimization, we should notice that all stages are treated as dependent processes (because of the situation presented in the example) i.e. there is the threat that tokens are allocated in the first stages in such a way, that it will essentially increase the total dislocation cost in further stages, hence, we must join the dislocation stages. Let us consider the variant when $d(i, j) = \text{const}$.

Table 1. Data structure and example of data flowing between stage 1, 2 and m

Stage 1	$t^{(0)}(i)$	To node 1	To node 2	...	To node n	$dt^{(1)}(i)$
From node 1	$t^{(0)}(1)$	0	$f^{(1)}(1, 2)$	(...)	$f^{(1)}(1, n)$	$dt^{(1)}(1)$
From node 2	$t^{(0)}(2)$	$f^{(1)}(2, 1)$	0	(...)	$f^{(1)}(2, n)$	$dt^{(1)}(2)$
...	(...)	(...)	(...)	(...)	(...)	(...)
From node n	$t^{(0)}(n)$	$f^{(1)}(n, 1)$	$f^{(1)}(n, 2)$	(...)	0	$dt^{(1)}(n)$
		$t^{(1)}(1)$	$t^{(1)}(2)$	(...)	$t^{(1)}(n)$	
Stage 2	$t^{(1)}(i)$	To node 1	To node 2	...	To node n	$dt^{(2)}(i)$
From node 1	$t^{(1)}(1)$	0	$f^{(2)}(1, 2)$	(...)	$f^{(2)}(1, n)$	$dt^{(2)}(1)$
From node 2	$t^{(1)}(2)$	$f^{(2)}(2, 1)$	0	(...)	$f^{(2)}(2, n)$	$dt^{(2)}(2)$
...	(...)	(...)	(...)	(...)	(...)	(...)
From node n	$t^{(1)}(n)$	$f^{(2)}(n, 1)$	$f^{(2)}(n, 2)$	(...)	0	$dt^{(2)}(n)$
		$t^{(1)}(1)$	$t^{(2)}(2)$	(...)	$t^{(2)}(n)$	
Stage m	$t^{(m-1)}(i)$	To node 1	To node 2	...	To node n	$dt^{(m)}(i)$
From node 1	$t^{(m-1)}(1)$	0	$f^{(m)}(1, 2)$	(...)	$f^{(m)}(1, n)$	$dt^{(m)}(1)$
From node 2	$t^{(m-1)}(2)$	$f^{(m)}(2, 1)$	0	(...)	$f^{(m)}(2, n)$	$dt^{(m)}(2)$
...	(...)	(...)	(...)	(...)	(...)	(...)
From node n	$t^{(m-1)}(n)$	$f^{(m)}(n, 1)$	$f^{(m)}(n, 2)$	(...)	0	$dt^{(m)}(n)$
		$t^{(m)}(1)$	$t^{(m)}(2)$	(...)	$t^{(m)}(n)$	

Table 2. Distances are stable $d(i, j) = d(j, i) = const$ for all stages (simplified variant)

Distances between nodes				
Nodes	1	2	(...)	n
1	0	$d(1, 2)$	(...)	$d(1, n)$
2	$d(2, 1)$	0	(...)	$d(2, n)$
(...)	(...)	(...)	(...)	5
n	$d(n, 1)$	$d(n, 2)$	(...)	0

First, we estimate the scale of token changes in all stages for particular nodes: $dp(i) = dt^{(m)}(i) - t^{(0)}(i)$; $dr(i) = |dt^{(m)}(i) - t^{(0)}(i)|$, $i = 1, 2, \dots, n$, ($dt^{(0)} = t^{(0)}$). Token transition is realized from the node with negative value $dp(i) > 0$ to the node with positive $dp(j) < 0$. The level of token transition is determined by expression: $\min\{ddr(i), ddr(j)\}$, where $ddr(i)$ - maximal (from among all stages), not yet realized, transition increments for i node and a fulfilled condition: $dr(i) \geq ddr(i)$. Real relative token increments will be calculated: $r(i) = \sum_{k=1}^m \sum_{j=1, j \neq i}^n f^{(k)}(i, j)$; $k = 1, 2, \dots, m$ ($r(i) \leq dr(i)$). Next, we order the nodes according to non-increasing changing values: $dr_{(1)} \geq dr_{(2)} \geq \dots \geq dr_{(n)}$, where the upper index shows the increment position in an ordered sequence. It refers to the sequence of node codes: $co(1), co(2), \dots, co(n)$. Then for every node we place them in a non-decreasing order in relation to other nodes distances:

$$\begin{array}{c}
 d_{(1)}(1) \leq d_{(2)}(1) \leq (\dots) \leq d_{(n-1)}(1) \\
 d_{(1)}(1) \leq d_{(2)}(1) \leq (\dots) \leq d_{(n-1)}(1) \\
 (\dots) \leq (\dots) \leq (\dots) \leq (\dots) \\
 d_{(1)}(n) \leq d_{(2)}(n) \leq (\dots) \leq d_{(n-1)}(n)
 \end{array}$$

where $d_{(j)}(i)$ - the distance from i node to node which is located on the j position in an ordered sequence. It is a corresponding sequences of node codes:

$cd_{(1)}(1), cd_{(2)}(1), (\dots), cd_{(n-1)}(1)$
$cd_{(1)}(2), cd_{(2)}(2), (\dots), cd_{(n-1)}(2)$
$(\dots), (\dots), (\dots), (\dots)$
$cd_{(1)}(n), cd_{(2)}(n), (\dots), cd_{(n-1)}(n)$

Let us consider the following approach:

1. Sequentially allocate tokens starting from the greatest total change estimated according to $t^{(0)}$ stage and the smallest distance, with regard to the direction of allocation (relative variant) [see Fig. 1].
2. Sequentially allocate tokens starting from greatest change in the stage with the maximal float and smallest distance with regards to the direction of allocation, choosing stages according to the maximal token change (increment variant).
3. Place the token changes in a non increasing order tokens changing $ddr^{(k)}(i) = |dt^{(k)}(i) - dt^{(k-1)}(i)|$ and $ddp^{(k)}(i) = dt^{(k)}(i) - dt^{(k-1)}(i)$ in all stages: $ddr_{(1)} \geq ddr_{(2)} \geq \dots \geq ddr_{(n*m)}$, where a low index means a position in an ordered sequence. It refers to the sequence of node codes: $cdo(1), cdo(2), \dots, cdo(n * m)$ [see Fig. 2].

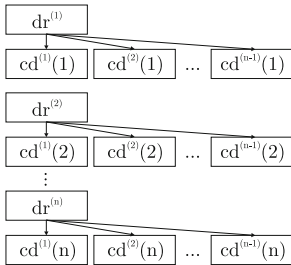


Fig. 1. Scheme of allocation tokens to nodes

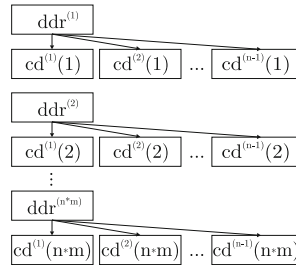


Fig. 2. Scheme of allocation tokens to nodes

We can create a simplified form of algorithm to describe two mentioned variants.

5.1 Relative Variant - An Algorithm for the Relative Approach

```

program optimization_relative;
procedure relative_inc_def; dr(i);dp(i),ddr(i);
procedure order_relative_inc;co(i);
procedure order_dist;cd(j)(i);
procedure max_singl_inc;cdo_max(i);
begin
    relative_inc_def;
    order_relative_inc
    
```

```

order_dist;
for i:=1 to n do
  for j:=1 to n do
    begin
      if ((dp(co(i)< 0)) and (dp(cd(j)(i)> 0))) then
        begin
          max_singl_inc (cdo_max(i));
          dp(co(i)):=dp(co(i))+ddr(cdo_max(i));
          dr(cd(j)(i)):=dr(cd(j)(i)- ddr(cdo_max(i));
          ddp(co(i)):=ddp(co(i))+ddr(cdo_max(i));
          ddr(cd(j)(i)):=ddr(cd(j)(i)- ddr(cdo_max(i))
        end;
      if ((dp(co(1)> 0)) and (dp(cd(j)(i)< 0)) then
        begin
          max_singl_inc (cdo_max(j));
          dp(co(i)):=dp(co(i)-ddr(cdo_max(j));
          dr(cd(j)(i)):=dr(cd(j)(i)+ ddr(cdo_max(j));
          ddp(co(i)):=ddp(co(i)-ddr(cdo_max(j));
          ddr(cd(j)(i)):=ddr(cd(j)(i)+ ddr(cdo_max(j))
        end;
    end;
  end;
end.

```

The "relative_inc_def" procedure estimates the values of all $dp(i)$, $dr(i)$ and their components $ddr^{(k)}(i)$ for all nodes (i), where $k = 1, 2, \dots, m$ is the stage number. The "order_relative_inc" procedure orders nodes according to predefined token changing values, creating a sequence $co(1), co(2), \dots, co(n)$. Procedure "order_dist" orders distances between the chosen i node and all other nodes and form a set of ordered distances $cd_{(1)}(i) \leq cd_{(2)}(i) \leq \dots \leq cd_{(n-1)}(i)$, $i = 1, 2, \dots, n$. The procedure "max_singl_inc" calculates the value of real transition as $ddr(cdo_max(i)) = \min\{dr(i), dr(j), \max_{k=1,2,\dots,m}\{ddr^{(k)}(i)\}\}$. That means the procedure calculates the maximal transition on the basis of ingredient increments and transition possibilities of the connected nodes: i - node with a maximal value of needs ($max\ dr(i) \rightarrow co(1)$) and j -node located nearest to the chosen i .

5.2 Increment Variant - An Algorithm for the Incremental Approach

```

program optimization_increment;
procedure inc_def; ddr(i);
procedure order_inc;cdo(i);
procedure order_dist;cd(j)(i);
procedure max_singl_inc;cdo_max(i);
begin
  relative_inc_def;

```

```

order_relative_inc
order_dist;
for i:=1 to n do
  for j:=1 to n do
    begin
      if ((ddp(cdo(i)< 0)) and (ddp(cd (j)(i)> 0)) then
        begin
          max_singl_inc (cdo_max(i));
          ddp(cdo(i):=ddp(cdo(i)+ddp(cdo_max(i));
          ddp(cd(j)(i):=ddp(cd(j)(i)- ddp(cdo_max(i))
        end;
      if ((dp(co(1)> 0)) and (dp(cd(j)(i)< 0)) then
        begin
          max_singl_inc (cdo_max(j));
          ddp(cdo(i):=ddp(cdo(i)-ddp(cdo_max(j));
          ddp(cd(j)(i):=ddp(cd(j)(i)+ ddp(cdo_max(j))
        end;
    end;
end.

```

The operation of the proposed optimization method exploiting cost criterion is best presented with the help of an example. Let us start with a table of data [see Table 3]. First, we calculate two kinds of token changes in relation to the initial state [see Table 4] and create a sequence of nodes according to $dr(i) : co(1) = 2; co(2) = 3; co(3) = 1; co(4) = 4$. And then for every node a sequence of neighbour nodes is formed according to the distances $cd_{(j)}(i)$ [see Table 5].

Now, for the first transition we choose nodes 2 and 1. Let us calculate the ingredient increments $ddr^{(k)}(i), ddp^{(k)}(i)$ [see Table 6]. First a transition of 4 tokens from node 3 to node 2 : $f^{(1)}(3, 2) = -4$, and from node 1 to 2 : $f^{(2)}(1, 2) = 1$ will be realized. The tables of the determined flows have the following shape: Table 7 in the first stage and Table 8 in the second stage. After correcting $dr^{(k)}(i), dp^{(k)}(i), ddr^{(k)}(i), ddp^{(k)}(i)$ in Table 9 the tables of flows present thyself as follows [see Tables 10 and 11]. And after subsequent corrections $dr^{(k)}(i), dp^{(k)}(i), ddr^{(k)}(i), ddp^{(k)}(i)$ we get Table 12. Then, we realize the following flows [see Tables 13 and 14] and finally the full array of optimal flows is in Table 15.

The total cost of network balancing determined by *relative variant* and is expressed by

$$c = |f^{(1)}(3, 1)| * d(3, 1) + |f^{(1)}(4, 1)| * d(4, 1) + |f^{(1)}(3, 2)| * d(3, 2) + |f^{(2)}(1, 2)| * d(1, 2) + |f^{(2)}(1, 3)| * d(1, 3) + |f^{(2)}(1, 4)| * d(1, 4) = 1 * 4, 2 + 2 * 11 + 4 * 2, 2 + 1 * 2, 2 + 2 * 4, 2 + 2 * 11 = 67, 6$$

Using the second algorithm *increment variant* we, for this data, obtain the same flow but in a different sequence. Therefore the total cost will be the same.

Table 3. Table of data

	$t^{(0)}(i)$	$dt^{(1)}(i)$	$dt^{(2)}(i)$	$d(i, 1)$	$d(i, 2)$	$d(i, 3)$	$d(i, 4)$
node 1	3	0	5	0	2,2	4,2	11
node 2	8	4	3	2,2	0	2,2	10
node 3	4	9	7	4,2	2,2	0	8,2
node 4	1	3	1	11	10	8,2	0

Table 4. Initial state

	$t^{(0)}(i)$	$dr(i)$	$dp(i)$
node 1	3	2	2
node 2	8	5	-5
node 3	4	3	3
node 4	1	0	0

Table 5. Sequences of neighbour nodes

	$cd_{(1)}(i)$	$cd_{(2)}(i)$	$cd_{(3)}(i)$
node 1	2	3	4
node 2	3	1	4
node 3	2	1	4
node 4	3	2	1

Table 6. Values of the increments

	$ddr^{(1)}(i)$	$ddp^{(1)}(i)$	$ddr^{(2)}(i)$	$ddp^{(2)}(i)$
node 1	3	-3	5	5
node 2	4	-4	1	-1
node 3	5	5	2	-2
node 4	2	2	2	-2

Table 7. Table of flows

	$f^{(1)}(i, 1)$	$f^{(1)}(i, 2)$	$f^{(1)}(i, 3)$	$f^{(1)}(i, 4)$
node 1				
node 2			4	
node 3		-4		
node 4				

Table 8. Table of flows

	$f^{(2)}(i, 1)$	$f^{(2)}(i, 2)$	$f^{(2)}(i, 3)$	$f^{(2)}(i, 4)$
node 1		-1		
node 2	1			
node 3				
node 4				

Table 9. Values of the increments

	$dr(i)$	$dp(i)$	$ddr^{(1)}(i)$	$ddp^{(1)}(i)$	$ddr^{(2)}(i)$	$ddp^{(2)}(i)$
node 1	1	1	3	-3	4	4
node 2	0	0	0	0	0	0
node 3	1	-1	1	1	2	-2
node 4	0	0	2	2	2	-2

Table 10. Table of flows

	$f^{(1)}(i, 1)$	$f^{(1)}(i, 2)$	$f^{(1)}(i, 3)$	$f^{(1)}(i, 4)$
node 1			1	
node 2				
node 3	-1			
node 4				

Table 11. Table of flows

	$f^{(1)}(i, 1)$	$f^{(1)}(i, 2)$	$f^{(1)}(i, 3)$	$f^{(1)}(i, 4)$
node 1				2
node 2				
node 3				
node 4	-2			

Table 12. Values of the increments

	$dr(i)$	$dp(i)$	$ddr^{(1)}(i)$	$ddp^{(1)}(i)$	$ddr^{(2)}(i)$	$ddp^{(2)}(i)$
node 1	4	-4	0	0	4	4
node 2	0	0	0	0	0	0
node 3	2	-2	0	0	2	-2
node 4	2	-2	0	0	2	-2

Table 13. Table of flows

	$f^{(2)}(i, 1)$	$f^{(2)}(i, 2)$	$f^{(2)}(i, 3)$	$f^{(2)}(i, 4)$
node 1			-2	
node 2				
node 3	2			
node 4				

Table 14. Table of flows

	$f^{(2)}(i, 1)$	$f^{(2)}(i, 2)$	$f^{(2)}(i, 3)$	$f^{(2)}(i, 4)$
node 1				-2
node 2				
node 3				
node 4	2			

Table 15. Table of all optimal flows

	$f^{(1)}(i, 1)$	$f^{(1)}(i, 2)$	$f^{(1)}(i, 3)$	$f^{(1)}(i, 4)$	$f^{(2)}(i, 1)$	$f^{(2)}(i, 2)$	$f^{(2)}(i, 3)$	$f^{(2)}(i, 4)$
node 1			1	2		-1	-2	-2
node 2			4		1			
node 3	-1	-4			2			
node 4	-2				2			

6 Conclusions and Future Work

Network balance problem is an important issue encountered mainly in the fields of logistics and computer networks. There were two centralised algorithms presented in this paper. The purpose of which is to compare the optimal results to the decisions made by a Multi-Agent System for network balance problem. As future work we plan to develop a Multi-Agent System for network balance problem and analyse it's effectiveness toward the presented here solutions and a classical distributed approach.

The two proposed algorithms possess the important advantage of being simplistic, so the implementation and the sequence of stages allows for in-depth analysis of the network and tokens at hand. It is based on a classical transporting approach with such a difference that in the analysed case we have a set of stages with mutually involved intermediate data. The advantage of using the proposed algorithms is that a set of optimal flows in all stages is available.

References

1. Fiol, M.A., Yebra, J.L.A., Alegre, I., Valero, M.: A discrete optimization problem in local networks and data alignment. *IEEE Trans. Comput. C* 36(6), 702–713 (1987)
2. Hong, B., Prasanna, V.K.: Maximum lifetime data sensing and extraction in energy constrained networked sensor systems. *J. Par. Distrib. Comput* 66, 566–577 (2006)
3. Piech, H.: Dependences and Configurations of Solution in Multi-criteria Optimization in Nets. In: *CCIS*, vol. 39, pp. 143–151. Springer, Heidelberg (2009)
4. Chiasserinia C.F., Rao R.R.: Stochastic battery discharge in portable communication devices. *IEEE Aerospace and Electronic Systems Magazine*, 41-45 (2000)
5. Escudero, M., Fiabrea, J., Morillo, P.: Fault-tolerant routing in double-loop networks. *Ars Combinatorial* 25A, 187–198 (1988)
6. Horton, M., et al.: Deployment ready multimode micropower wireless sensor networks for intrusion detection, classification and tracking (C3I) Technologies for Homeland Defense and Low Enforcement, 290–295 (2002)
7. Guan, D.J.: Routing a vehicle of capacity greater than one. *Discrete Applied Mathematics* 81(1), 41–57 (1998)
8. Czubak, A., Wojtanowski, J.: Lifespan-Aware Routing for Wireless Sensor Networks. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6071, pp. 72–81. Springer, Heidelberg (2010)

A New Cluster-based Instance Selection Algorithm

Ireneusz Czarnowski and Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{irek,pj}@am.gdynia.pl

Abstract. The main contribution of the paper is proposing and evaluating, through the computational experiment, an agent-based population learning algorithm generating a representative training dataset of the required size. The proposed approach is based on the assumption that prototypes are selected from clusters. Thus, the number of clusters produced has a direct influence on the size of the reduced dataset. Agents within an A-Team execute various local search procedures and cooperate to find-out a solution to the instance reduction problem aiming at obtaining a compact representation of the dataset. Computational experiment has confirmed that the proposed algorithm is competitive to other approaches.

Keywords: data reduction, instance selection, clustering, machine learning, optimization, population learning algorithm, A-Team.

1 Introduction

The paper focuses on data reduction. Data reduction in the supervised machine learning aims at deciding which instances from the training set should be retained for further use during the learning process [11]. Data reduction can be achieved through selecting instances and/or through selecting features.

Data reduction is considered especially useful as a mean to increasing effectiveness of the learning process when the available datasets are large, such as those encountered in data mining, text categorization, financial forecasting, mining of multimedia databases and meteorological, financial, industrial and science repositories, analysing huge string data like genome sequences, Web documents and log data analysis, mining of photos and videos, or information filtering in E-commerce [5, 11, 17, 20]. Finding a small set of representative instances, also called patterns, prototypes or reference vectors, for large datasets can result in producing a classification model superior to one constructed from the whole massive dataset. Besides, such an approach may help to avoid working on the whole original dataset all the time [21]. It is also obvious that removing some instances from the training set reduces time and memory complexity of the learning process [12, 19].

Data reduction is also considered as an important step towards increasing effectiveness of the learning process when the available datasets are large or distributed and when the access to data is limited and costly from the commercial point of view. In the distributed data mining (DDM) a simple approach is to move all of the data

from distributed sites to a central site, merging the data and building a single global learning model. Unfortunately, moving all data into a centralized location can be very time consuming, costly, or may not be feasible due to some restrictions [12, 13, 15].

Selecting the relevant data from distributed locations and then moving only the local patterns can eliminate or reduce the restrictions on a communication bandwidth, reduce the cost of data shipping, and speed up the distributed learning process [8]. The important distributed data mining problem is to establish a reasonable upper bound on the size of the dataset needed for an efficient analysis [18].

The paper deals with instance selection. The main contribution of the paper is proposing and evaluating through computational experiment an agent-based population learning algorithm generating a representative dataset of the required size. The proposed approach is based on the assumption that prototypes are selected from clusters. Thus, the number of clusters produced has direct influence on the size of the reduced dataset. The instance selection method proposed in this paper is an extension of the approach introduced in [6]. In [6] the instance selection was carried in two stages. At the first stage clusters were generated using the similarity coefficient as the criterion for the instance selection procedure. The final number of clusters was obtained by merging the initially generated clusters aiming at minimizing the proximity measures. The merging was repeated several times to obtain the required data compression rate and several merging procedure were considered. In the presented approach the initial clusters are generated in the same way. Next, when the initially generated clusters does not assure the required data compression rate the clusters are merged under the learning process executed by the team of agents.

The goal of the paper is to show through computational experiment that the proposed cluster-based instance selection approach assures the required data compression rate and that it can be competitive to its first version as presented in [6], as well as to several other data reduction algorithms known from the literature. To validate the approach, a number of computational experiment runs have been carried out. Performance of the proposed algorithm has been evaluated using several benchmark datasets from UCI repository [1].

The paper is organized as follows. The agent-based instance selection algorithm and its features are presented in Section 2. Section 3 provides details on the computational experiment setup and discusses its results. Finally, the last section contains conclusions and suggestions for future research.

2 An Approach to the Cluster-based Instance Selection

Since instance selection belongs to the class of computationally difficult combinatorial optimization problems [8], to solve its instances it is proposed to apply one of the population-based metaheuristics known as the population-learning algorithm proposed originally in [10]. In the proposed implementation the optimization and improvement procedures are executed by the set of agents cooperating and exchanging information within an asynchronous team of agents (A-Team).

The section contains an overview of the dedicated agent-based architecture used, including its main features, and gives a detailed description of the proposed agent-based population learning algorithms for the cluster-based instance selection.

2.1 A-Team Concept

The A-Team architecture has been proposed as a problem-solving paradigm that can be easily used to design and implement the proposed population learning algorithm carrying-out the instance reduction tasks. The A-Team concept was originally introduced in [16]. The idea of the A-Team was motivated by several approaches like blackboard systems and evolutionary algorithms, which have proven to be able to help successfully solving some difficult combinatorial optimization problems. Within an A-Team agents achieve an implicit cooperation by sharing a population of solutions to the problem to be solved. Such a population of solutions is an equivalent of the population of individuals known from the evolutionary algorithms.

In the discussed population-based multi-agent approach multiple agents search for the best solution using local search heuristics and population based methods. The best solution is selected from the population of potential solutions which are kept in the common memory. Specialized agents try to improve solutions from the common memory by changing values of the decision variables. All agents can work asynchronously and in parallel. During their work agents cooperate to construct, find and improve solutions which are read from the shared common memory. Their interactions provide for the required adaptation capabilities and for the evolution of the population of potential solutions.

The main functionality of the agent-based population learning approach includes organizing and conducting the process of searching for the best solution. It involves a sequence of the following steps:

- Generation of the initial population of solutions to be stored in the common memory.
- Activation of optimizing agents which apply solution improvement algorithms to solutions drawn from the common memory and store them back after the attempted improvement applying some user defined replacement strategy.
- Continuation of the reading-improving-replacing cycle until a stopping criterion is met. Such a criterion can be defined either or both as a predefined number of iterations or a limiting time period during which optimizing agents do not manage to improve the current best solution. After computation has been stopped the best solution achieved so far is accepted as final.

To implement the agent-based population learning algorithm one has to set and define the following:

- Solution representation format
- Initial population of individuals
- Fitness function
- Improvement procedures
- Replacement strategy implemented for managing the population of individuals.

More information on the population learning algorithm with optimization procedures implemented as agents within an asynchronous team of agents (A-Team) can be found in [2]. In [2] also several its implementations are described.

2.2 Agent-Based Population Learning Algorithm for Instance Selection

This paper proposes an A-Team in which agents execute the improvement procedure and cooperate in a manner described in the preceding subsection with a view to solve instances of the data reduction problem. The approach is called ABIS (Agent-Based Instance Selection).

The basic assumptions behind the proposed agent-based instance selection approach are following:

- Instances are selected from clusters of instances. Clusters are constructed separately from the set of training instances with the identical class label.
- Prototype instances are selected from clusters through the population-based search carried out by the optimizing agents.
- Clusters are induced in two stages. At the first stage the initial clusters are produced using the procedure based on the similarity coefficient. The instances are grouped into clusters according to their similarity coefficient calculated as in [5]. The clusters induced at the first stage contain instances with identical similarity coefficient. The main feature of the discussed clustering procedure is that the number of clusters is determined by the value of the similarity coefficient (see, for example, [5]). The second stage involves merging of clusters obtained at the initial stage. Clusters are merged through the population-based search. The merging is carried-out in case when the number of clusters obtained at the first stage does not assure the required data compression rate. Such case means that the number of clusters obtained at the cluster initialization stage is greater than the upper bound of the number of clusters set by the user. Hence, the solution produced using the similarity coefficient only, is not feasible and must be further improved at the second stage.
- Initially, potential solutions are generated through randomly selecting exactly one single instance from each of the considered clusters (either merged or not merged).
- Each solution from the population is evaluated and the value of its fitness is calculated. The evaluation is carried out by estimating classification accuracy of the classifier, which is constructed using instances (prototypes) indicated by the solution as the training dataset.

A feasible solution is represented by two data structures: a string and a binary square matrix of bits. A string contains numbers (numeric labels) of instances selected as prototypes. A total length of the string is equal to the number of clusters of potential reference instances. The following example explains the notation used. Assume that the number of instances in the original data set is 15, numbered from 1 to 15. Let the considered string be:

$$s = [4,7,12],$$

thus, the number of clusters of potential reference instances equals 3. Instance number 4 from the training dataset has been selected to represent the first cluster as the reference vector. Similarly, instance number 7 has been selected as the reference vector for the second cluster, and instance number 12 as the reference vector for the third one.

A matrix of bits $M = [m_{ij}]_{n \times n}$, where n is the initial number of clusters, denotes whether or not clusters, induced at the cluster initialization stage, have been merged with a view to comply with the constraint on the number of reference vector allowed. The element $m_{ij}=1$, which lies in the i -th row and the j -th column of the matrix M , denotes that clusters i and j are merged. In addition the discussed matrix has the following properties:

- (1) $\sum_{ij} m_{ij} = k$, where k is the upper bound on the allowed number of clusters and hence, number of the reference vectors.
- (2) $\forall_i \sum_j m_{ij} = 1$.
- (3) $\forall_{i,j:i=j}$ the element m_{ij} of a matrix M is an artificial “missing value”.

The following example explains the above notation. Let the number of the required clusters is equal to 3 and the number of clusters induced at the cluster initialization stage is equal to 6. Thus the 6-by-6 matrix should be considered. Let the considered matrix be:

$$M = \begin{bmatrix} - & 1 & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 1 & 0 \\ 0 & 1 & 0 & - & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & 0 & 0 & - \end{bmatrix}$$

where $m_{12}=1$ denotes that the first cluster is merged with the second, $m_{35}=1$ denotes that the third cluster is merged with the fifth, and $m_{42}=1$ denotes that the fourth cluster is merged with the second.

Initially, for each individual in the population of solutions, the corresponding matrix M conforming with properties (1) – (3) is generated randomly. To solve the cluster-based instance selection problem, the following two groups of optimizing agents carrying out different improvement procedures have been implemented:

- The first group includes the improvement procedures for instance selection as it was proposed in [4]. These procedures are: the local search with the tabu list for instance selection and the simple local search. The first procedure - local search with the tabu list for instance selection, modifies a solution by replacing a randomly selected reference instance with some other randomly chosen reference instance thus far not included within the improved solution. The modification takes place providing the replacement move is not on the tabu list. After the modification, the move is placed on the tabu list and remains there for a given number of iterations. The second procedure - simple local search modifies the current solution either by removing the randomly selected reference instance or by adding some other randomly selected reference instance thus far not included within the improved solution.

- The second group consists of the optimizing agents responsible for merging clusters. Agents execute a simple local search procedure. The procedure modifies the current solution changing the composition of clusters through transferring a single reference vector from randomly selected merged cluster into another randomly selected cluster in each iteration. The merging procedure is shown as Algorithm 1.

Algorithm 1. Local search for cluster merging

Input: $M_{d:d=1,\dots,m}$ - parts of the solution encoded as square matrices for clusters containing instances belonging to a single class. The dimensions of each matrix is $n_d \times n_d$, where m - denotes the number of the decision classes and n_d is the initial number of clusters for the class d .

Output: $M'_{d:d=1,\dots,m}$ - the improved solutions encoded as a matrix.

1. Set d by drawing it at random from $\{1, 2, \dots, m\}$;
2. Choose randomly two elements from the matrix M_d referred to as m_{ij} and m_{gh} (where $i, j, g, h=1, \dots, n_d$) such that $m_{ij}=1$ and $m_{gh}=0$;
3. Replace the values between m_{ij} and m_{gh} thus producing M'_d satisfying conditions (1) and (2);
4. If (M'_d is better than M_d) then goto 6;
5. If (!terminating condition) then goto 1;
6. Return M'_d ;

In each of the above cases the modified solution replaces the current one if it is evaluated as a better one using the classification accuracy as the criterion. If, during the search, an agent successfully improves the solution then it stops and the improved solution is stored in the common memory. Otherwise, agents stop searching for an improvement after having completed the prescribed number of iterations.

The proposed A-Team uses a simple replacement strategy. Each optimizing agent receives a solution drawn at random from the population of solutions (individuals). The solution returned by optimizing agent is merged with the current population replacing the current worst solution.

3 Computational Experiment

To validate the proposed approach it has been decided to carry out the computational experiment. The experiment aimed at answering the following two basic questions:

- Does the ABIS assure appropriate compression rate?
- Does the ABIS perform better than other data reduction algorithms?

To validate the proposed approach several benchmark classification problems have been solved. Datasets for each problem including Cleveland heart disease, credit approval, Wisconsin breast cancer, Ionosphere, Hepatitis, Diabetes and Sonar have been

obtained from the UCI Machine Learning Repository [1]. Characteristics of these datasets are shown in Table 1.

Each benchmarking problem has been solved 30 times, and the experiment plan involved 3 repetitions of the 10-cross-validation scheme. The reported values of the quality measure have been averaged over all runs. The quality measure in all cases was the correct classification ratio. In the 10-cross-validation scheme, for each fold, the training dataset was reduced using the proposed approach.

Table 1. Datasets used in the reported experiment

Dataset	Number of instances	Number of attributes	Number of classes	The best reported classification accuracy
Heart	303	13	2	90.0% [7]
Sonar	208	60	2	97.1% [1]
Australian credit (ACredit)	690	15	2	86.9% [1]
German credit (GCredit)	1000	20	2	77.47%[9]
Cancer	699	9	2	97.5% [1]
Ionosphere	351	34	2	94.9% [1]
Hepatitis	155	19	2	87.13%[9]
Diabetes	768	8	2	77.34%[9]

The proposed algorithm has been run three times, with the upper bound on the number of the selected prototypes set to $t\%$ of the number of instances in the original dataset, where during the experiment t was equal to 10, 15 and 20, respectively.

During the experiment population size for each investigated A-Team architecture was set to 20. The process of searching for the best solution has been stopped either after 100 iterations or earlier in case there has been no improvement of the current best solution for one minute of computation. Values of these parameters were set arbitrarily.

The proposed A-Team has been implemented using the middleware environment called JABAT [2], based on JAVA code and built using JADE (Java Agent Development Framework) [3].

Classification accuracy of the classifier obtained using the proposed approach (i.e. using the set of prototypes, found by selecting instances and removing irrelevant attributes) has been compared with:

- Results obtained by machine classification without data reduction, i.e. on full, non-reduced dataset.
- Results obtained using the set of prototypes produced through selection based on the k -means clustering (In this case at the first stage the k -means clustering has been implemented and next, from thus obtained clusters, the prototypes have been selected using the agent-based population learning algorithm as in [10]).
- Results obtained using the first version of the proposed algorithm (in this case at the second stage the initially generated clusters are merged using the average linkage cluster merging strategy – ALP [6]).

Generalization accuracy has been used as the performance criterion. The learning tool used was the C4.5 algorithm [14]. The experiment results are shown in Table 2. The ranking of the compared approaches is shown in Fig. 1, where horizontal axis represents the mean relative difference between the mean accuracies of the best method and the given method. The above discussed results are averaged over all experiments.

To evaluate the performance of compared algorithms the Friedman's non-parametric test has been used. The test aimed at deciding the following hypotheses:

- H_0 – null hypothesis: instance selection algorithms are statistically equally effective regardless of the kind of the problem.
- H_1 – alternative hypothesis: not all instance selection algorithms are equally effective.

The analysis has been carried out at the significance level of 0.05. The respective value χ^2 statistics with 10 algorithms and 8 instances of the considered problems is 32 and the value of χ^2 distribution is equal to 16.9 for 9 degree of freedom. Analysis of the experiment results shows that for the population of the classification accuracy observations the null hypothesis should be rejected. Thus, not all algorithms are equally effective regardless of the kind of problem which instances are being solved. In Fig. 1 average weights for each instance selection algorithm are shown.

Table 2. Accuracy of the classification results (%)

	Heart	Cancer	ACredit	Sonar	Hep.	GCredit	Ino.	Diab.
C 4.5*	77.89	94.57	84.93	74.04	83.87	70.5	91.45	73.82
ALP ($t=10\%$)	84.64	96.30	86.66	78.32	81.34	72.21	88.47	78.3
ALP ($t=15\%$)	87.84	98.20	88.14	76.62	83.84	74.04	87.13	76.6
ALP ($t=20\%$)	86.52	97.62	89.20	74.62	84.03	73.1	89.46	77.21
k -means ($t=10\%$)	85.67	94.43	88.99	54.34	75.67	68.01	84.54	72.1
k -means ($t=15\%$)	87.67	95.09	90.14	59.48	75.32	69.8	83.32	73.7
k -means ($t=20\%$)	86.00	96.14	90.29	72.17	76.06	68.4	82.32	74.53
ABIS ($t=10\%$)	84.0	97.30	88.55	80.77	79.6	77.2	90.59	80.35
ABIS ($t=15\%$)	86.33	98.86	90.69	80.77	84.03	78.1	91.45	80.61
ABIS ($t=20\%$)	87.26	97.58	89.56	81.73	82.04	78.7	92.01	80.61

*- non-reduced data set.

From Fig. 1, one can observe that the best results have been obtained by the ABIS algorithm. The ABIS algorithm assures the required size of the reduced dataset and is competitive to the earlier version – ALP. The worst result have been produced by the k -means-based approach. It should be also noted that the cluster-based approach produced very good results as compared with case when the classifier is induced using original, non-reduced dataset.

The experiment results also show that the agent-based search is a suitable tool for solving complex optimization problems including cluster-based instance selection.

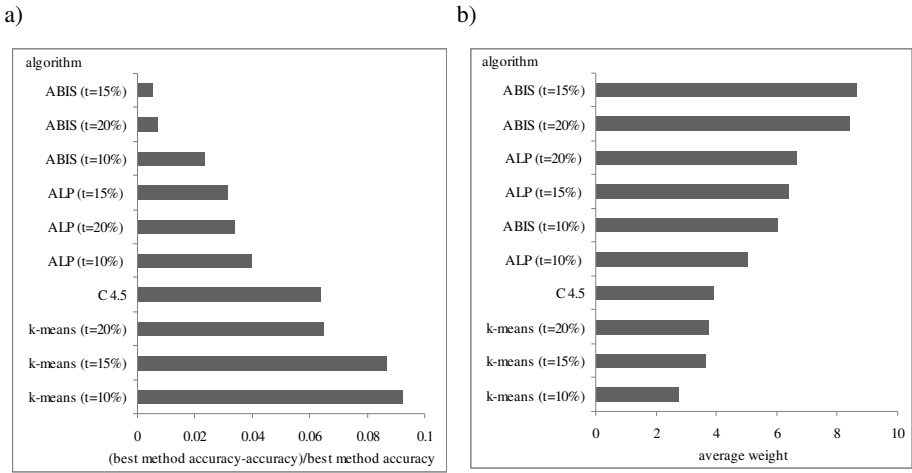


Fig. 1. Ranking of the instance selection algorithms (a) and the average Friedman test weights (b)

4 Conclusions

The paper proposes an approach to instance selection based on cluster integration. The main property of the proposed algorithm is that the instances are selected from clusters, where the number of clusters determines the final size of the reduced dataset. The instances are selected by agent-based population learning algorithm, which role is also to produce the appropriate number of clusters.

The main contribution of the paper is proposing and evaluating through computational experiment a new cluster-based instance selection algorithm. In the reported computational experiment the proposed algorithm outperformed other cluster-based algorithms.

Future research will focus on establishing rules for finding an optimal configuration of the A-Team carrying out the data reduction task.

Acknowledgments. This research has been supported by the Polish Ministry of Science and Higher Education with grant for years 2010-2013.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. School of Information and Computer Science. University of California, Irvine (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: e-JABAT - An Implementation of the Web-Based A-Team. In: Nguyen, N.T., Jain, I.C. (eds.) Intelligent Agents in the Evolution of Web and Applications. SCI, vol. 167, pp. 57–86. Springer, Heidelberg (2009)

3. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. *Exp.* 3(3), 6–20 (2003)
4. Czarnowski, I., Jędrzejowicz, P.: An Approach to Data Reduction and Integrated Machine Classification. *New Generation Computing* 28(1), 21–40 (2010)
5. Czarnowski, I., Jędrzejowicz, P.: An Approach to Instance Reduction in Supervised Learning. In: Coenen, F., Preece, A., Macintosh, A. (eds.) *Research and Development in Intelligent Systems XX*, pp. 267–282. Springer, London (2004)
6. Czarnowski, I., Jędrzejowicz, P.: Cluster Integration for the Cluster-Based Instance Selection. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) *ICCCI 2010. LNCS*, vol. 6421, pp. 353–362. Springer, Heidelberg (2010)
7. Datasets used for classification: comparison of results. directory of data sets, <http://www.is.umk.pl/projects/datasets.html> (accessed September 1, 2009)
8. Hamo, Y., Markovitch, S.: The COMPSET Algorithm for Subset Selection. In: *Proceedings of The Nineteenth International Joint Conference for Artificial Intelligence*, Edinburgh, Scotland, pp. 728–733 (2005)
9. Jędrzejowicz, J., Jędrzejowicz, P.: Cellular GEP-Induced Classifiers. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) *ICCCI 2010. LNCS*, vol. 6421, pp. 343–352. Springer, Heidelberg (2010)
10. Jędrzejowicz, P.: Social Learning Algorithm as a Tool for Solving Some Difficult Scheduling Problems. *Foundation of Computing and Decision Sciences* 24, 51–66 (1999)
11. Kim, S.-W., Oommen, B.J.: A Brief Taxonomy and Ranking of Creative Prototype Reduction Schemes. *Pattern Analysis Application* 6, 232–244 (2003)
12. Klusch, M., Lodi, S., Moro, G.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) *Intelligent Information Agents. LNCS (LNAI)*, vol. 2586, pp. 104–122. Springer, Heidelberg (2003)
13. Krishnaswamy, S., Zaslavsky, A., Loke, S.W.: Techniques for Estimating the Computation and Communication Costs of Distributed Data Mining. In: Sloot, P.M.A., Tan, C.J.K., Dongarra, J., Hoekstra, A.G. (eds.) *ICCS-ComputSci 2002. LNCS*, vol. 2329, pp. 603–612. Springer, Heidelberg (2002)
14. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, SanMateo (1993)
15. Silva, J., Giannella, C., Bhargava, R., Kargupta, H., Klusch, M.: Distributed Data Mining and Agents. *Engineering Applications of Artificial Intelligence Journal* 18, 791–807 (2005)
16. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Co-operation Schemes for Autonomous, Computer-Based Agents. Technical Report EDRC 18-59-96, Carnegie Mellon University, Pittsburgh (1996)
17. Uno, T.: Multi-sorting Algorithm for Finding Pairs of Similar Short Substrings from Large-scale String Data. *Knowledge and Information Systems* (2009); doi: 10.1007/s10115-009-0271-6
18. Vucetic, S., Obradovic, Z.: Performance Controlled Data Reduction for Knowledge Discovery in Distributed Databases. In: *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 29–39 (2000)
19. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-based Learning Algorithm. *Machine Learning* 33(3), 257–286 (2000)
20. Yu, K., Xiaowei, X., Ester, M., Kriegel, H.-P.: Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach. *Knowledge and Information Systems* 5(2), 201–224 (2004)
21. Zhu, X., Wu, X.: Scalable Representative Instance Selection and Ranking. In: *IEEE Proceedings of the 18th International Conference on Pattern Recognition*, vol. 3, pp. 352–355 (2006)

An Agent-Based Approach to the Dynamic Price Problem

Yuri Chizhov

Department of Modelling and Simulation
Riga Technical University
1 Kalku Street, Riga LV-1658, Latvia
Jurijs.Cizovs@rtu.lv

Abstract. This paper focuses on the agent realisation of the Markov Decision Process in the dynamic price problem. The use of MDP is caused by the properties of the trade system under consideration. Due to exploiting data mining tools, time series processing, clustering and other operations, the obtained interaction architecture has turned to be overloaded. Taking into account particular properties of the considered trade system as well as its model, which is MDP-based, this paper suggests a novel specific multiagent technique for MDP system realisation.

Keywords: Artificial Intelligent Agents, Markov Decision Processes, Dynamic Pricing Policy.

1 Introduction

A Markov Decision Process (MDP) is a method of dynamic optimisation task realisation. A task in the MDP statement has several solution algorithms. In its turn, intelligent agents are a way to represent an algorithm that in this case aims to solve the task evolving over time. According to the agent paradigm, the algorithm under consideration has to possess such properties as ability to solve the task stated, learning (adaptation) ability, ability to operate autonomously, purposefulness etc.

Specific features of the task of decision making about the price value for the manufactured products (see Section 1) cause using MDP for modelling the process of price correction. In this task a real trade system is considered; here an object of MDP operation is sales data that represent multidimensional time series. To transform them into a structure compatible with an MDP model, it is necessary to exploit data mining for working with the time series. To construct a more effective and a more visual architecture, it is proposed to represent an MDP as an agent system. It should be noted that an MDP from the first foresees an agent representation of the system. This is due to several basic properties of MDP, including process management over time, the presence of the “target” and the presence of effect and perception tools, constitute the key properties of agent systems (see Section 2).

Due to a broad spectrum of tasks solved by agent systems, several techniques for implementing agent MDPs exist: for example, it may be implemented in the conditions when the correlation between agents is either partial or is not present at all, or might be developed as a coordinator in multiagent systems and so on. However, the

lack of publications on the development of MDP-based agent systems applied to real economic tasks is observed nowadays. Section 3 addresses agent MDP realisation in the task of dynamic pricing policy (DPP). A technique for environment determination is described that is based on the “points” of its perception by an agent and of agent impact, which solves conflicts in determining the environment in the case of heterogeneous agents and in the case of the distributed environment structure when the source of the data perceived by the agent is not the receiver of its effectors impacts.

2 Markov Decision Process in Dynamic Price Policy

Economic effectiveness of trade and industrial enterprises can be achieved by using various tools one of which is reasonable pricing policy. The aim of price correction could be to maximise long-term profit, to “survive” under the conditions of tight competition and to avoid human factor mistakes in fixing the price at the large number of wholesalers and diversity of products. The task is to formulate pricing policy for the given market system and to work out the mechanism for its construction.

The diversity of task parameters yields different kinds of systems, e.g., (a) systems with one seller or a market with many sellers, (b) systems having automated information exchange between all participants of the market, (c) systems with different types of trade: wholesale trade, retail trade, mixed trade etc. Due to that, there is no one universal approach to the task statement, policy representation technique and policy construction mechanism that would fit all the systems. Instead, many particular techniques for describing the task of dynamic prices are available.

In this paper the DPP task is a system with one seller and many wholesale buyers each of which purchases deliberate subsets of goods. The prices are defined within the dimensional space Product-Buyer-Time. Taking into account the specific features of real industrial enterprise, these peculiarities of the DPP task are assumed:

- the task is of the form *First-degree price discrimination*, accordingly, “*the buyer-product*” system is considered [1];
- measurements «The Products» and «The Buyer» are described by hundreds of values, which makes using a simple search for finding a decision more complicated;
- the process of decision making on the price value is represented by a finite number of states S . The environment is non-deterministic but fully observable;
- the state S is a time series – development of the prices and sales volumes over time for a certain product and a wholesale buyer;
- the impact on the system is represented by price correction for a certain product and a wholesale buyer;
- the process of price correction is partly controlled because some uncontrolled parameters are present (e.g., sales volume);
- certain kinds of product may have infinite planning horizon, which means that the end of production is not planned yet;
- a finite planning horizon is also possible. Pricing policies at the finite and infinite planning horizons will be different;
- the data about sales, prices, products and buyers are stored in the relational database.

Several models for calculating dynamic prices can be found in literature. Some researchers formulate this task as optimisation task [2]. This paper suggests using MDPs due to the following reasons.

Firstly, MDP enables model construction. Besides factors like cost, sales-demand ratio and marketing actions, the price of a product depends on many other factors which cannot be fully taken into account by trade and industrial software. In this case, historic sales data can be used where price dynamics represents response to one or another factor. Therefore the decision regarding the new price will be based on price model that is based on actual sales figures. The price model is a representation of the current state (or states window with the length Δt) as a changing value of the price for the studied object of sale. Application of price model means introducing and developing price policy that will determine the consequence of solutions that will help to achieve the desired situation. Price policy should take into account not only the costs of the product itself, but also a range of relevant factors, such as: who is the buyer (to which buyers' group does it relate), volume of the sale for one transaction, the time of the year and possible national holidays, state of the competitors and external-economic factors.

Secondly, this task is of dynamic nature. From time to time the sales manager draws up a solution regarding price for any specific product for wholesale buyers. Sales volume, competitiveness and other indicators depend on this solution. Therefore, we have a dynamic system with a specific state at any given moment. Transition from one state to the other is executed by changing parameters, for example by changing the price. Such task is attributed to the problems of Dynamic Programming (DP), and therefore, can be solved by decomposing it into a consequence of steps (solutions). In the work [3] DPP-task is examined from the DP point of view.

One of the instruments for solving such problems is MDP. As a type of stochastic process it is widely used for researching and modelling development possibility of a specific system in the long-term. MDPs introduce two benefits: they take into account the long-term effects of each recommendation and the expected value of each recommendation. Works [4, 5, 6] use MDP with hidden states for stock markets forecasting tasks, decision making on buying or selling, forecasting prices for air transportation market, modelling the market of electric power that depend on a wide range of complex economic influences.

Formally MDP is defined as a sequence (1):

$$M = \langle S, A, P, R \rangle \quad (1)$$

Where S – a finite set of states, A – a finite set of actions, P – probability that the action a , taken in the state s at the time t , leads to the state s' at the time $t+1$: $P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$, R – instant reward acquired after transition to the state s' from the state s with the transition probability $P_a(s, s')$, thus, $R_a(s, s') = \mathcal{R}(s_{t+1} = s' | s_t = s, a_t = a)$. The task of MDP is to find the policy of actions that will maximise the full sums of rewards in the potentially infinite time-line:

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1}) \quad (2)$$

where $a_t = \pi(s_t)$, γ -discount factor $0 < \gamma < 1$ (usually is closer to 1). As the policy of the function $\pi(s)$ that determines which action to take being in the state s , the MDP process together with the policy acts as a Markov chain.

As any other algorithm, MDP is demanding regarding the expression of the task's source data. Toy problems like *Grid world*, *Mountain Car*, *Taxi problem* and others are initially "compatible" with MDP. Difficulties arise when MDP is used for real tasks in economics, management etc. as there the data structure firstly corresponds to criteria of the initial task (e.g. data processing). When applying MDP, the parameters of the system that form the state of the process should be determined, as well as the set of actions, the function or the matrix of transactions and rewards. It is also difficult to determine the interaction of MDP with given tasks. In real tasks, for example, in trade-economic tasks, data is stored in database and it cannot be "loaded" into MDP as data is constantly renewed and is very extensive and cannot be processed fast.

Therefore, it is necessary to apply the algorithm to the data and not the data to the algorithm. This kind of approach is provided by agent systems where the agent is embedded into the environment of the task and is functioning inside it by influencing it and receiving signals from it. Further we will examine different types of agents and will choose the most suitable for solving DPP tasks via MDP.

3 Agent Systems

Modern computer systems, network solutions, mobile technologies, global positioning systems etc. create technological basis to develop more complex software. If a system has a combination of characteristics like autonomy, adaptability, functionality in real time, interaction with similar systems, mobility etc. it is attributed to agent systems.

Let's use these definitions of an agent: «*An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future*» [7] and the definition proposed by IBM's Intelligent Agent Strategy white paper: "*Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires*".

Table 1. Agent properties

Property	Description
Reactive (sensing and acting)	responds in a timely fashion to changes in the environment
Autonomous	exercises control over its own actions
Goal-oriented (pro-active purposeful)	does not simply act in response to the environment
Temporally continuous	is a continuously running process
Communicative	communicates with other agents and/or people
Learning (adaptive)	changes its behaviour based on its previous experience
Mobile	able to transport itself from one environment to another

Both definitions are based on endowing a system with particular properties. Depending on the set of properties, the following main classes are highlighted: programmable agents (reactive agents, reflexive agents [8]), learning agents and planning agents [9]. Table 1 shows properties that the agent of any class can possess (drawn up by [7]).

Agents do not represent any algorithms or functions, but they do represent the particular form of how the algorithms are expressed to solve the task. Such an approach is the basis of the AOSE paradigm (Agent-Oriented Software Engineering) [10, 11] that is the development of Object-Oriented Programming.

The concept of the agent-based approach stipulates that the agents are functioning inside the environment [8, 7]. From our point of view, the concept of the environment is the most important difference between programs and agents. In the case of program, the task is loaded into the program as a flow or as files, but in the case of agent approach an agent as a “solver” is immersed into the task and perceives it with sensors fully or by fragments. When “solver” is immersed into the environment, its perception and influence on the task becomes external regarding the “solver” itself. The separation of the “solver” and the task requires data transfer channels from the environment to the agent and from the agent to the environment. This role is played by sensors and effectors correspondingly. Agent-environment interaction is widely described in appropriate literature, for example [12].

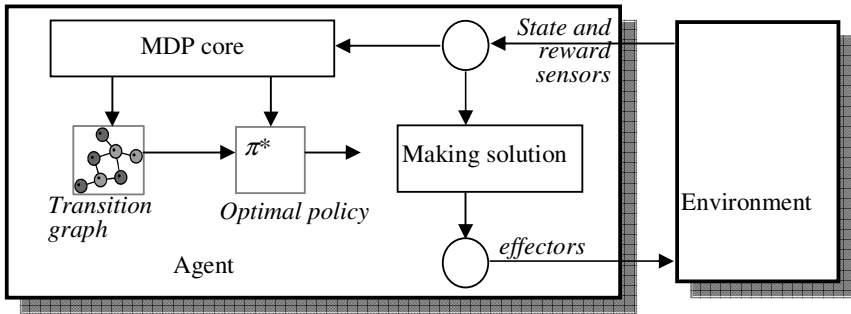


Fig. 1. Agent structure in case of MDP/RL logic

Considering the separation of the agent from the environment, an agent needs to build the model of the environment inside its own space in order to make decisions. Frequently, environment model is expressed via a knowledge base. The knowledge base allows an agent to develop local action as an answer to incoming signal. The process of forming and correcting (renewing) knowledge base is the agent learning process and is realised by algorithms of artificial intelligence.

On the basis of common model of learning agents [8], Fig. 1 shows the structure of the agent that is based on MDP. The graph of MDP transitions is the model of the environment. Based on the transitions graph algorithm, Value Iteration or Policy Iteration develops optimal actions policy π^* by which the agent is guided when developing a decision. The fact that the transition graph that directly influences the

decision is subjected to settings provides the opportunity for agent learning. This represents the simplest way of agent expression of MDP.

According to [8], the described agent relates to the type of agents that operate on the basis of utility as it has utility function that displays current situation as real number that indicates the extent of agent's "satisfaction". Therefore it is possible to search for the state with maximal value. Agents can solve multicriteria tasks, satisfy conflicting goals (for example, speed and safety of driving a car) as well as find alternative solutions that correspond to the given level of utility.

4 Dynamic Price Policy Task Representation by Agent-Based MDP

Let's examine agent approach of MDP realisation in DPP-task. Expressing MDP via agent approach allows simplifying architecture thanks to the properties of agent systems (see Table 1). Autonomy, learning, goal-orientation, reactivity and other properties become obligatory for program subsystem.

4.1 Single-Agent MDP

To realise single-agent approach, external MDP module is interpreted as software agent with regard to economic system. Agent input (reception) is entries of new sales in a database. Influence of the agent is the decision about price correction that cannot be sent directly to the system, but is reported to the sales manager. In such case, the manager can only be called a part of the environment. Agent relates to the class of learning agents that act on the basis of utility as utility concept underlies MDP. Utility is calculated by searching current state in the transition model. Learning process of the agent is accomplished thanks to functions employed in processing new data that is constantly provided. Acquired regularities are included into transitions model.

The main drawback of the single-agent approach is functional overload of the agent. The database cannot be direct environment for Markov Decision Process. The data have to be transformed in a manner so as to form a set of states, provide state transitions model, as well as foresee reinforcement and goal states. Therefore Data Mining should be engaged. As a result, an agent is provided with functions that form time series, find regularities, create and store sales behaviour profiles etc. Overloading imply difficulties in determining agents environment. Moreover, it becomes difficult to attract external means to solve intermediate tasks, for example clustering of time series.

4.2 Multiagent MDP

In this work, the multiagent approach does not change the studied system of one seller and multiple wholesale buyers. The approach is used to distribute functions of one agent among several agents. In such case every sub-task of one agent becomes the sole task (goal) of every separate agent.

The number of agents in multiagent system is determined by the number of functions that are employed in the price correction subsystem. Functions that transform

initial data into structures corresponding to the requirements of Markov Decision Processes, are realised by three reactive agents (perform data reflection from one form into another) and create “infrastructure” for signals provision from external environment to the decision making agent in the form that satisfies MDP. MDP-agent can receive the signal in the form of state and reward and develop decision regarding price correction. Reactive agents are interface between MDP-agent and database and they create visibility of direct MDP-agent interaction with the environment.

The obtained multiagent system also suffers from insufficient clarity of environment definition as definition of environment for MDP-agent is irrelevant for reactive agents. Reactive agents have completely different input and output signals, aspire different goals which place these agents into completely different environment.

4.3 Polymorph Environment in Multiagent System

Classical representation of multiagent environment as a common environment in which a set of agents are functioning, is not enough to fully and precisely describe their interaction in modern tasks. Let's examine two following prerequisites.

First prerequisite: direct interaction with the task. In model tasks of *Grid world*, *Mountain Car* and others, the agents interact with the environment directly – sensors perceive parameters of the environment and form the state, effectors influence these parameters. In such a view none of the DPP-task agents interact with the environment as they do in model tasks. It is known that for the agent the environment is the “provider” of the task and the “field” for solutions. But in the case of DPP-task the definition that the environment is what the agent interacts with [13] should be elaborated, as reactive agents do not have direct influence on the database and MDP-agent does not interact physically with data on sales. Therefore, conflict situation arises when agents do not interact directly with the environment that provides the task but they solve it.

Second prerequisite. Several sources [13, 8] define environment as *everything that lies outside the agent*. For modelling tasks, for example for *Vampus world* it is admissible. In fact, if a multiagent system and agents have conflicting goals, then for every agent $a_i \in A$ other agents can be interpreted as part of the environment. But in this case the environment will be different for every agent a_i . For example, in the task of “*Vampus labyrinth*” nothing will disturb the agent from viewing *Vampus* itself as dynamic parameter. Assuming that *Vampus* is an agent his “victim” can be interpreted as dynamic parameter of the environment. But even in modelling task such approach already brings uncertainty regarding environment frames.

Thus, there is necessity to re-evaluate the structure of multiagent environment because of the indirect interaction of the agents with the environment and because of difficulties in description of what is the environment for every agent.

Let's assume that environment E consists of set of components (including agents, database and operator). Every agent a_i by fact interacts with its own perception of the environment E_i . To strictly determine frames of the environment E_i we will repulse from interface points (sensors and effects) of the agent a_i with the environment E . In classical definition (for example [8]) of agent interaction with the environment, the environment is a monolith object that at the same time is the source of sensor data for the object and is also the receiver of effectors' influences (see Fig. 1). In other words both interaction points have common contact point with the environment.

Dividing environment into interaction points (of all participants) allows us to view the environment as multicomponent environment expressed as a set (3).

$$E = \{e_1, e_2, \dots, e_n\} \tag{3}$$

Then for a particular agent a_i its own perception of the environment is a subset $E_i \in E$, which includes only those components e_j of the environment E with which the agent interacts.

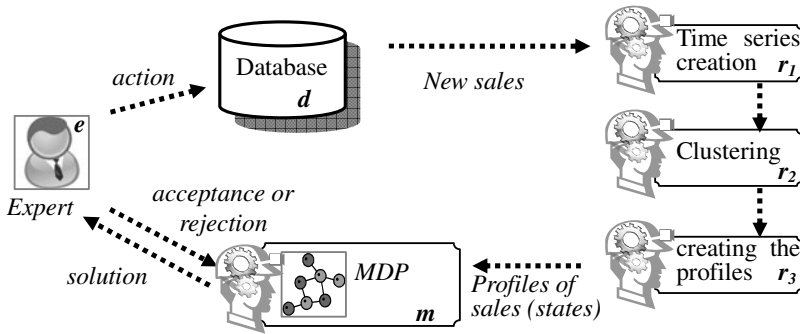


Fig. 2. Multiagent environment with defined components

For the other agent the environment will be represented as a set of different components. The property of the environment to “represent” different contents for different agents we will define as polymorphous.

In the case of dynamic price policy the use of polymorphic (multicomponent) environment allows one to resolve the problem of defining environment for different types of agents that pursue different goals. In fact non reactive agent knows nothing about goals of MDP-agent. Similarly, MDP-agent perceives signals of reactive agents, but he sees them as the environment and not as “collaborators”.

Let’s use the environment that has polymorphic property to define environment view of every agent in the DPP-task. Let’s define components of the environment: database (d), reactive agents (r_1, r_2, r_3), MDP-agent (m), expert (e). Fig. 2 depicts the environment $E = \{d, r_1, r_2, r_3, m, e\}$.

The decomposition of the environment into components and definition of components interaction (data transmission channels) allows one to describe environment E view and sub-task for each agent. Table 2 displays corresponding subsets $E()$. The given table describes location of the agent in the environment via its interaction points for every task. The environment is divided into logical and physical levels. A logical level is determined by the task that has to be solved. A physical is determined by a particular perception of the environment $E()$. The relation lies in the fact that the environment on the physical level through corresponding components provides an agent with the access to the environment of logical level. In other words, the physical level represents interface points which provide access to the task.

Table 2. Each agent's particular view of the environment

Task	Sensors	Effectors	Subsets of E	Agent
Time series building	Sales records	Time series	$E(r_1) = \{d, r_2\}$	Reactive agent (r_1)
Clustering	Time series	Clusters	$E(r_2) = \{r_1, r_3\}$	Reactive agent (r_2)
Regularities searching	Clusters	Profiles of sales behaviour	$E(r_3) = \{r_2, m\}$	Reactive agent (r_3)
Dynamic price policy	Profiles of sales behaviour	Price correction decision	$E(m) = \{r_3, e\}$	MDP-agent (m)

As can be seen from Table 2, physical environment for MDP-agent is a subset $E(m)$, including the expert and reactive agent r_3 which provides behaviour profiles. From our point of view, it might be difficult to say that such combination is an environment, but for MDP-agent it is defined by the fact that through this combination the agent interacts with the task. Reactive agent r_3 and sales manager are the interface, creating the “visibility” of the external environment for MDP-agent.

Therefore, we acquired the definition of the particular environment perception of every agent based on the source of incoming signals and receiver of outgoing signals from the agent's effectors. Using this approach there are no difficulties regarding the definition of the environment in multiagent system when the agents pursue different goals. Moreover, the task of possible inclusion of additional agent will be narrowed down to defining “location” of these agents in the system that is defining interface points. Particular representation of the environment allows one to divide it into physical and logical environments and thus resolves the conflict in the definition of the agent task if the object of the task is not reachable directly.

5 Conclusions and Future Works

The tasks characterized by sequential data processing require that multiagent system environment definition is extended. This paper has demonstrated several techniques for representing an MDP process through a multiagent architecture. The need to transform real environment data into the MDP structure makes it necessary to employ sequential data processing. The use of agents has provided the *Environment - Data Mining - MDP* architecture with some additional properties possessive in agent systems. To avoid conflicts in determining the environment, a polymorphic property is introduced, which enables particular environment representation and the possibility of incorporating additional agents without causing any conflicts in determining the environment and tasks of each agent.

References

1. Varian, H.R.: Differential Pricing and Efficiency. First Monday 1(2) (1996)
2. Narahari, Y., Raju, C.V.L., Ravikumar, K., Shah, S.: Dynamic pricing models for electronic business. *Sādhanā* 30(2 & 3), 231–256 (2005); Printed in India

3. Chizhov, Y.A., Zmanovska, T.P., Borisov, A.N.: Temporal Data Mining for Identifying Customer Behaviour Patterns. In: Perner, P. (ed.) ICDM 2009. LNCS, vol. 5633, pp. 22–32. Springer, Heidelberg (2009)
4. Sunderejan, R., Kumar, P.N.A., Krishnan, B.K.N., Dinesh, R., Kartikeyan, S., Sridhar, C.: Stock market trend prediction using Markov models. In: Proceedings of RTCSP 2009, pp. 285–289. Electron, Coimbatore (2009)
5. Rafiul, M. H., Nath, B.: Stock Market Forecasting Using Hidden Markov Model: A New Approach. In: Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications, pp. 192–196. IEEE Computer Society, Washington, DC, USA (2005)
6. Wang, Y., Sheblé, G.B.: Modeling electricity markets with hidden Markov model. *Electric Power Systems Research* 76(6-7), 445–451 (2006)
7. Franklin, S., Graesser, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Tambe, M., Müller, J., Wooldridge, M.J. (eds.) IJCAI-WS 1995 and ATAL 1995. LNCS, vol. 1037, Springer, Heidelberg (1996)
8. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
9. Jacobs, S.: *Applying ReadyLog to Agent Programming in Interactive Computer Games*. Diplomarbeit, Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen (2005)
10. Lind, J.: Issues in Agent-Oriented Software Engineering. In: Ciancarini, P., Wooldridge, M.J. (eds.) AOSE 2000. LNCS, vol. 1957, pp. 45–58. Springer, Heidelberg (2001)
11. Lind, J.: Agent Oriented Software Engineering, <http://www.agentlab.de/aose.html>
12. Matthew, E.T., Stone, P.: Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research* 10, 1633–1684 (2009)
13. Sutton, R.S., Barto, A.G.: *Reinforcement Learning*. MIT Press, Cambridge (1998)

Parallel Cooperating A-Teams Solving Instances of the Euclidean Planar Travelling Salesman Problem

Piotr Jędrzejowicz and Izabela Wierzbowska

Department of Information Systems, Gdynia Maritime Academy,
Morska 81-87, 81-225 Gdynia, Poland
{pj, iza}@am.gdynia.pl

Abstract. The paper proposes an architecture for solving difficult combinatorial optimization problems through cooperation of several A-Teams working in parallel. The idea extends JABAT environment through integrating the team of asynchronous agent paradigm with the island-based genetic algorithm concept. The proposed architecture has been tested on instances of the Euclidean planar travelling salesman problem. Computational experiment has confirmed that the proposed architecture is competitive in terms of the quality of solutions in comparison with using traditional, non-cooperating A-Teams.

Keywords: Parallel A-Teams, agent cooperation, combinatorial optimization, Euclidean planar travelling salesman problem.

1 Introduction

The techniques used to solve difficult combinatorial optimization problems have evolved from constructive algorithms to local search techniques, and finally to population-based algorithms. In recent years, technological advances have enabled development of various parallel and distributed versions of the population based methods. At the same time, as a result of convergence of many technologies within computer science such as object-oriented programming, distributed computing and artificial life, the agent technology has emerged. An agent is understood here as any piece of software that is designed to use intelligence to automatically carry out an assigned task, mainly retrieving, processing and delivering information.

Paradigms of the population-based methods and multiple agent systems have been during mid nineties integrated within the concept of the asynchronous team of agents (A-Team). A-Team is a multi agent architecture, which has been proposed in several papers of S.N. Talukdar and co-authors [7], [8], [9], [10].

The middleware platforms supporting implementation of A-Teams are represented by the JADE-Based A-Team environment (JABAT). Its different versions were proposed in [1], [4] and [5]. The JABAT middleware was built with the use of JADE (Java Agent Development Framework), a software framework proposed by TILAB [2]. JABAT complies with the requirements of the next generation A-Teams which are portable, scalable and in conformity with the FIPA standards. To solve a single

task (i.e. a single problem instance) JABAT uses a population of solutions that are improved by optimizing agents which represent different optimization algorithms. In traditional A-Teams agents work in parallel and independently and cooperate only indirectly using a common memory containing population of solutions. The paper proposes an architecture for solving difficult combinatorial optimization problems through cooperation of several A-Teams working in parallel. The idea extends JABAT environment through integrating the team of asynchronous agent paradigm with the island-based genetic algorithm concept first introduced in [3]. In the resulting Team of A-Teams (*TA-Teams*) two levels of cooperation are present. Cooperation at the lower level involves reading solutions from the common memory and storing them back in this memory after an attempted improvement. Cooperation at the upper level involves communication, that is information exchange, between cooperating A-Teams.

It is expected, following the idea of the island-based genetic algorithms, that information exchange between a number of parallel populations may help to escape from the local optima. Thus, the original A-Team architecture has been extended into the *TA-Teams* architecture through addition of the following features:

- Several A-Teams, working in parallel, are engaged in solving the same instance of the given problem. Each A-Team maintains own common memory where a number of the appropriately encoded problem solutions (individuals) are stored.
- A-Teams within a *TA-Teams* communicate through cyclically exchanging some solutions.

The paper is constructed as follows: Section 2 describes the *TA-Teams-EPTSP* which is an implementation of the specialized Team of A-Teams designed to solve instances of the Euclidean planar travelling salesman problem. The *TA-Teams-EPTSP* has been implemented using JABAT middleware environment. Section 3 gives details of the validating experiment settings and Section 4 contains a discussion of the experiment results. Finally some conclusions and suggestions for future research are drawn.

2 Team of A-Teams Implementation

2.1 Euclidean Planar Travelling Salesman Problem Formulation

Without losing a generality, the concept of *TA-Teams* is illustrated using, as an example, one of the best known difficult combinatorial optimization problems, the Euclidean planar travelling salesman problem (EPTSP). The discussed problem is a particular case of the general TSP. EPTSP can be formulated as follows: given n cities (points) in the plane and their Euclidean distances, the problem is to find the shortest TSP-tour, i.e. a closed path visiting each of the n cities exactly once.

2.2 A-Teams within *TA-Teams*

JABAT middleware environment can be used to implement A-Teams producing solutions to optimization problems using a set of optimizing agents, each representing an

improvement algorithm. Such an algorithm receives a solution and attempts to improve it. Afterwards, successful or not, the result is send back to where it came from. The process of solving a single task (that is an instance of the problem at hand) consists of several steps. At first the initial population of solutions is generated and stored in the common memory. Individuals forming the initial population are, at the following computation stages, improved by independently acting agents, each executing an improvement algorithm. Different improvement algorithms executed by different agents supposedly increase chances for reaching the global optimum. After a number of reading, improving and storing back cycles, when the stopping criterion is met, the best solution in the population is taken as the final result.

A typical JABAT implementation allows for running a number of A-Teams in parallel providing the required computational resources are available. Such A-Teams do not communicate nor cooperate, even indirectly. Unfortunately, as it has been confirmed by the computational experiment discussed in Section 4, advantages of constructing and running several A-Teams in parallel and solving the same task independently are not significant.

In the field of evolutionary computations many methods have been proposed and used to control diversity, prevent convergence and to distribute individuals over different areas of the search space. Among them one can mention distributed population models. Island based evolutionary computations belong to the distributed population models class. The idea is to carry out the evolutionary process within subpopulations before migrating some individuals to other islands and then continuing the process in cycles involving evolutionary processes and migrations (see, for example [11,12]).

The idea behind distributed population-based models and especially island-based evolutionary algorithms, has been implemented in the *TA-Teams-EPTSP*. The above architecture allows a number of A-Teams to work on solving the same task independently, exploring different regions of the search space. The added process of communicating between A-Teams makes it possible to exchange some solutions between common memories maintained by each of the A-Teams with a view to prevent premature convergence and assure diversity of individuals.

TA-Teams-EPTSP provides facility to define the process of migration that is the process of exchanging solutions between independent common memories. This process is supervised by a specialized agent called *MigrationManager* and defined by the following:

- *Migration size* - the number of individuals sent in one cycle from the common memory of one A-Team to the common memory of another A-Team.
- *Migration frequency* - length of time between migrations.
- *Migration topology* – specification of the communication channel between A-Teams. It specifies to which partner A-Team any given A-Team from *TA-Teams* will be periodically transmitting one or more of solutions from its own common memory. In this paper we consider a *TA-Teams* based on the ring architecture, which implies that each from A-Teams receives communication from one adjacent A-Team and sends communication to another adjacent A-Team.

- *Migration policy* – rules defining which solutions migrate and how they are added into the target population. In this paper we consider *best-worst* policy, in which the best solution is taken from the source population and replaces the worst solution in the target population.

2.3 Working Strategies

The process of solving a single task in JABAT by a single A-Team is defined by a working strategy understood as a set of rules responsible for the process of maintaining the common memory. The common memory contains a population of solutions called individuals. All individuals are feasible solutions of the instance to be solved.

A-Teams in *TA-Teams-EPTSP* follow the working strategy in which:

- All individuals in the initial population of solutions are generated randomly.
- Selection of individuals for improvement is a random move.
- Returning individual replaces the first found worse individual. If a worse individual can not be found within a certain number of reviews (review is understood as a search for the worse individual after an improved solution is returned) then the worst individual in the common memory is replaced by the randomly generated one representing a feasible solution.
- The computation time of a single A-Team is defined by the *no improvement time gap*. If in this time gap no improvement of the current best solution has been achieved the A-Team stops computations. Then all other A-Teams solving the same task stop as well, regardless of recent improvements in their best solutions.

The overall best result from common memories of all A-Teams in *TA-Teams* is taken as the final solution found for the task.

2.4 Optimizing Agents

Among several types of agents performing administration, maintenance, communication and optimization tasks which are standardized in JABAT, only optimizing agents directly influence the quality of individuals (i.e. solutions) stored in the common memory. Individuals from the common memory are selected in accordance with the current working strategy and forwarded for improvement to the independently running optimizing agents. Such an agent attempts to improve the received solution and, subsequently, resends it to the common memory.

Each agent is assigned to a specific A-Team and is working on improving solutions from its common memory. This is different from earlier implementations of JABAT, where agents could cooperate with a number of populations stored in different common memories, providing they all contained solutions of the same problem but, not necessarily the same instance.

In the reported *TA-Teams-EPTSP* there are five types of optimizing agents. Each of them attempts to improve the received solution executing utmost 100 local search moves. The first encountered improvement stops this search. Local search procedures used by the optimizing agents are as follows:

- *Simple exchange* (Ex2) – deletes two random edges from the input solution thus breaking the tour under improvement into two disconnected paths and reconnects them in the other possible way, reversing one path.
- *Triple exchange* (Ex3) – deletes three random edges from the input solution thus breaking the tour under improvement into three disconnected paths. These paths can be reconnected in two other possible ways. Algorithm chooses the best of the resulting tours.
- *Recombination1* (R1) – there are two input solutions. A subpath from one of them is randomly selected. In the next step it is supplemented with edges from the second solution. If this happens to be impossible the procedure constructs an edge connecting endpoint of the sub path with the closest point in the second input solution not yet in the resulting tour.
- *Recombination2* (R2) – there are two input solutions. A sub path from one of them is randomly selected. In the next step it is supplemented by the remaining points in a sequence taken from the second solution.
- *Mutation* (M) – two randomly selected points from the input solution are directly connected. This subpath is supplemented with edges as in R1. If this happens to be impossible the procedure constructs an edge connecting endpoint of the subpath with the closest remaining point from the initial solution.

3 Computational Experiment Settings

To evaluate the *TA-Teams-EPTSP* performance computational experiment has been carried out. Instances used in the reported experiment come from TSPLIB – a library of sample instances for the TSP [6]. The following instances have been used: pr76, pr107, pr144 and pr299 with 76, 107, 144 and 299 cities, respectively.

In each experiment run, for each A-Team, 5 optimizing agents have been employed, one of each type described in subsection 2.4. The remaining settings included:

- Maximum number of the allowed reviews upon a returning individual – 5.
- *Common memory size* in each of the A-Teams – 3, 5, 7 or 9 solutions.
- *Number of A-Teams* in *TA-Teams-EPTSP* - 3, 5 or 7.
- *No improvement time gap* – 1 or 2 minutes.
- *Migration frequency* – 0.3, 0.5 ... 2 minutes.
- *Migration size* - single solution.

Experiment has been carried out on the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core with 12 MB L3 cache processors with Mellanox InfiniBand interconnections with 10Gb/s bandwidth.

TA-Teams-EPTSP likewise JABAT, as derived from JADE, makes it possible to create agents containers on different machines and connecting them to the main platform. Then agents may migrate from the main platform to these containers. In each of the reported experiments 5 nodes of the cluster have been used - one for the main platform and four for the optimising agents to migrate to.

4 Computational Experiment Results

A number of experiments has been carried out for each instance and different parameters including common memory size, number of A-Teams or migration frequency. The results - in term of relative computation error - have been averaged over not less than ten runs. Computation errors have been calculated in relation to the optimum results known from the TSPLIB.

In the first series of experiments the impact of implementing the distributed population-based approach has been assessed. It has been done by comparing results obtained from *TA-Teams-EPTSP* used with the following architectures:

1. Single A-Team.
2. 5 A-Teams without communication nor migration feature.
3. Fully functional *TA-Teams-EPTSP* consisting of 5 A-Teams with migration every half minute.

In each A-Team *common memory size* was set to 7. On average, the computation error produced with the use of the fully functional *TA-Teams-EPTSP* is smaller than average errors generated in other considered cases, as it can be seen in Fig. 1. In accordance with the intuition, the worst performer has been the single A-Team. This has been true for all considered problem instances (tasks).

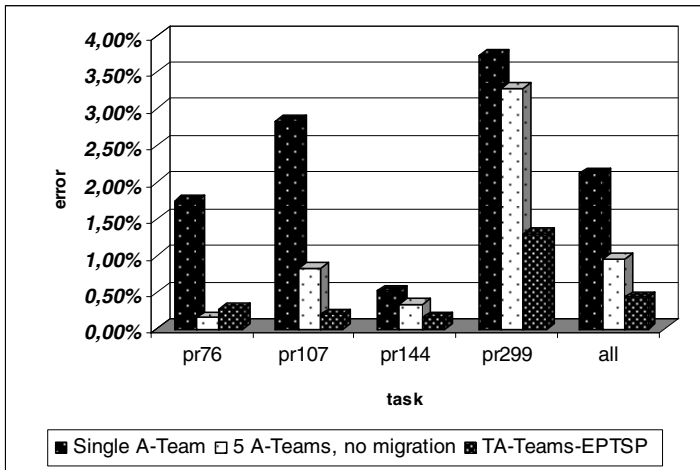


Fig. 1. Average computation errors for different architectures

In Fig. 1 the computation error was averaged over different settings of *no improvement time gap*, *migration frequency* and the number of solutions in each A-Team (population size). However, the same remains true for each particular setting. For example Fig. 2 presents average computation error for *no improvement time gap* 2, *migration frequency* 0.5 and *common memory size* 7.

Fig. 3 presents the percentage of experiments in which the optimal value has been found for the investigated architectures for smaller tasks and *no improvement time*

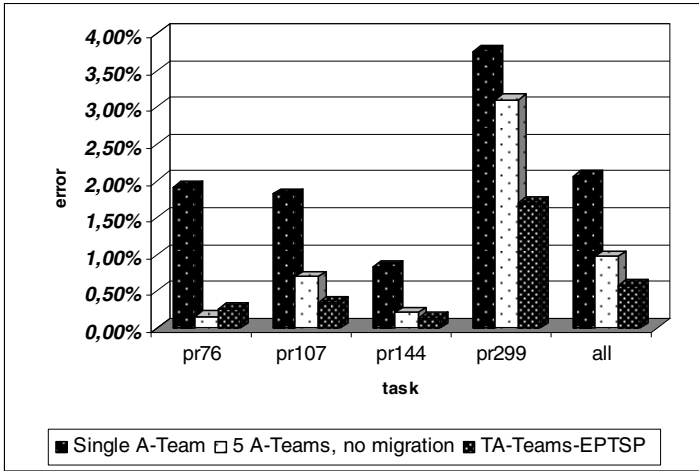


Fig. 2. Average computation errors for different architectures with *no improvement time gap* set to 2, *migration frequency* set to 0.5 and *common memory size* set to 7

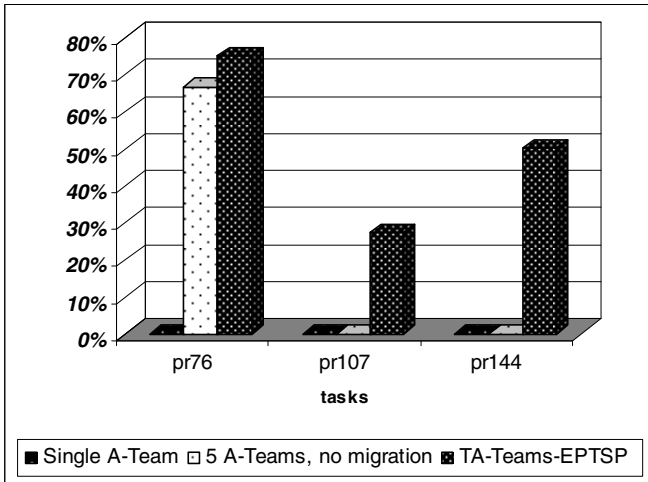


Fig. 3. The percentage of experiments in which the optimal value has been found

gap set to 2. Single A-Team has never found the optimal value. Again, solving the task using fully functional *TA-Teams-EPTSP* yields the optimal solution most often.

When *TA-Teams-EPTSP* with full functionality is used with the given migration topology and size, the parameters that may influence the results include: number and size of A-Teams in the *TA-Teams* and the migration frequency. Figures 4-6 show how these parameters affect computation errors. Data shown in the figures has been averaged over all runs of the experiment. It can be seen that increasing the number of A-Teams produces the most noticeable improvement, but only up to a point.

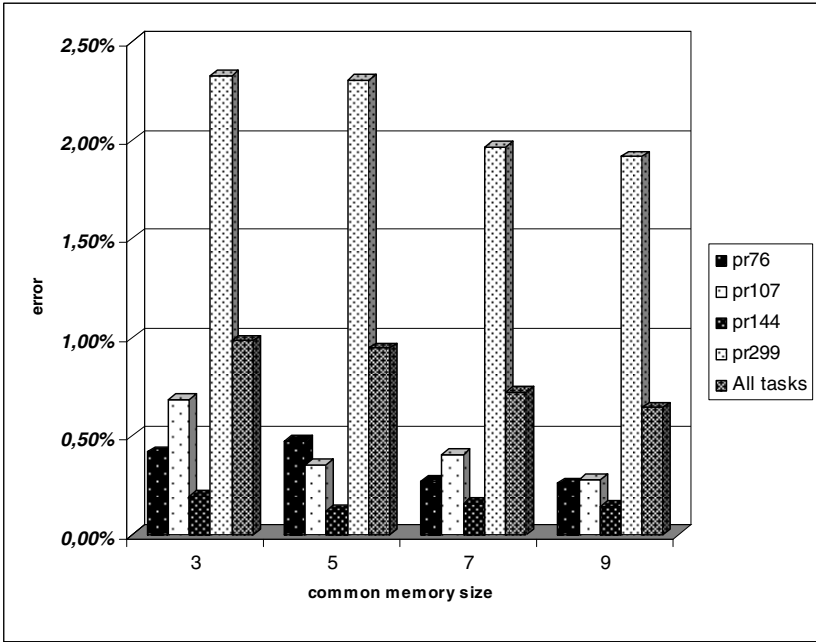


Fig. 4. Average computation errors for different common memory sizes within the *TA-Teams-EPTSP*

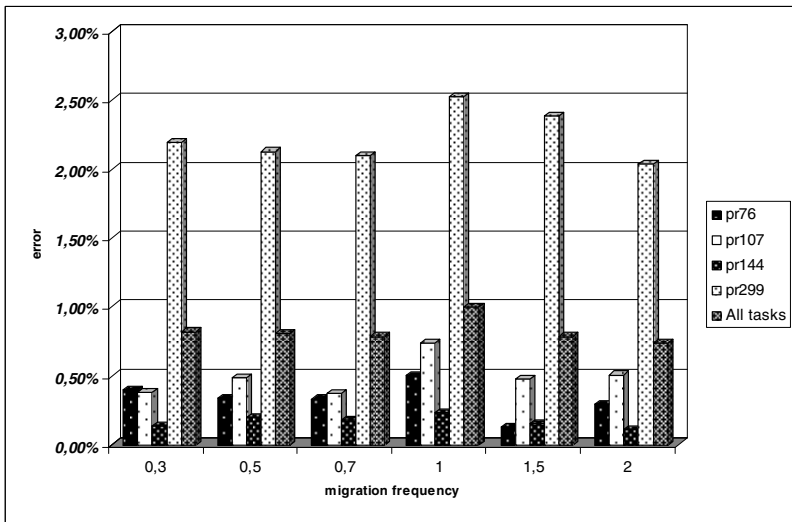


Fig. 5. Average computation errors for different migration frequencies within the *TA-Teams-EPTSP*

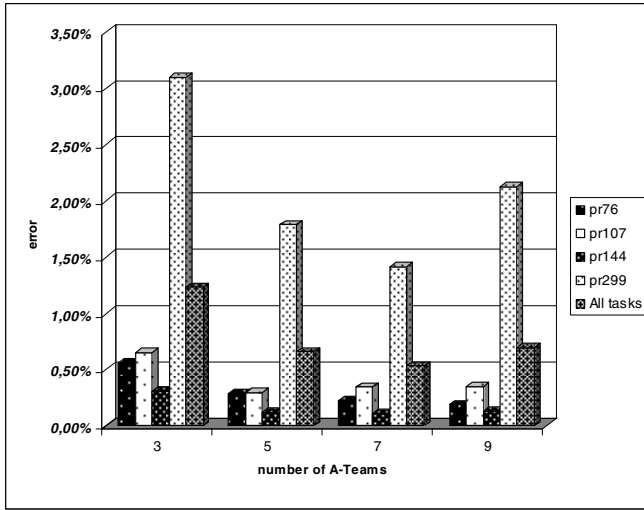


Fig. 6. Average computation errors for different numbers of A-Teams in *TA-Teams-EPTSP*

5 Conclusions

The presented research has confirmed that integrating the distributed evolutionary concept, and especially the island based evolutionary algorithm with the A-Team paradigm might result in achieving a noticeable improvement in the quality of the computation results. In case of the investigated Euclidean planar travelling salesman problem, where A-Teams have been constructed using relatively low quality optimizing agents using simple local search algorithms, the discussed effect has, in general, appeared in all the considered cases.

The range of parameters used in the experiments has been insufficient to draw further conclusions as to the choice of parameters in construction of the *TA-Teams* constructed from heterogeneous optimizing agents with a view to achieving best results. Future research will focus on evaluating effects of a wider set of parameters and solving more test data, which might be helpful in identifying more general observations.

Acknowledgments. Calculations have been performed in the Academic Computer Centre TASK in Gdańsk. The research has been supported by the Ministry of Science and Higher Education no. N N519 576438 for years 2010-2013.

References

1. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak, E., Wierzbowska, I.: JADE-Based A-Team as a Tool for Implementing Population-Based Algorithms. In: Chen, Y., Abraham, A. (eds.) *Intelligent Systems Design and Applications, IDSA*, Jinan Shandong China, pp. 144–149. IEEE, Los Alamos (2006)
2. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. *Exp.* 3(3), 6–20 (2003)

3. Cohoon, J.P., Hegde, S.U., Martin, W.N., Richards, D.: Punctuated Equilibria: a Parallel Genetic Algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 148–154. Lawrence Erlbaum Associates, Hillsdale (1987)
4. Czarnowski, I., Jędrzejowicz, P., Wierzbowska, I.: A-Team Middleware on a Cluster. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, pp. 764–772. Springer, Heidelberg (2009)
5. Jędrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
6. Library of sample instances for the TSP,
<http://comopt.ifi.uniheidelberg.de/software/TSPLIB95>
7. Talukdar, S.N.: Collaboration Rules for Autonomous Software Agents. *Decision Support Systems* 24, 269–278 (1999)
8. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous Agents. *Journal of Heuristics* 4(4), 295–332 (1998)
9. Talukdar, S.N., Pyo, S.S., Giras, T.: Asynchronous Procedures for Parallel Processing. *IEEE Trans. on PAS* 102(11), 3652–3659 (1983)
10. Talukdar, S.N., de Souza, P., Murthy, S.: Organizations for Computer-Based Agents. *Engineering Intelligent Systems* 1(2), 56–69 (1993)
11. Tanese, R.: Distributed genetic algorithms. In: Schaffer, J. (ed.) Proceedings of the Third International Conference on Genetic Algorithms, pp. 434–439. Morgan Kaufmann, San Mateo (1989)
12. Whitley, D., Rana, S., Heckendorn, R.B.: The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. *Journal of Computing and Information Technology* 7, 33–47 (1998)

A-Team for Solving MRCPSP/max Problem

Piotr Jędrzejowicz and Ewa Ratajczak-Ropel

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{pj,ewra}@am.gdynia.pl

Abstract. In this paper the E-JABAT-based A-Team architecture for solving multi-mode resource-constrained project scheduling problem with minimal and maximal time lags (MRCPSP/max) is proposed and experimentally validated. MRCPSP/max, also known as MRCPSP-GPR (MRCPSP with generalised precedence relations), belongs to the NP-hard problem class. To solve this problem an asynchronous team of agents implemented using JABAT middleware has been proposed. Three kinds of optimization agent has been used. Computational experiment involves evaluation of optimization agent performance within the A-Team. The paper contains the MRCPSP/max problem formulation, description of the proposed architecture for solving the problem instances, description of optimization algorithms and the discussion of the computational experiment results.

Keywords: project scheduling, multi-mode resource-constrained project scheduling, MRCPSP, optimization, agent, agent system, A-Team.

1 Introduction

The paper proposes an agent-based approach to solving instances of the MRCPSP/max, known also in the literature as the MRCPSP-GPR problem. MRCPSP stands for the Multi-mode Resource-Constrained Project Scheduling Problem, max or GPR is used to describe precedence relations as minimal and maximal time lags, also called Generalised Precedence Relations (GPR) or temporal constraints or time windows. MRCPSP/max has attracted a lot of attention and many exact and heuristic algorithms have been proposed for solving it (see for example [11], [4], [5], [9], [3]).

MRCPSP/max is a generalisation of RCPSP/max and thus it is NP-hard [2]. The approaches to solve this problem produce either approximate solutions or can be applied for solving instances of the limited size. Hence, searching for more effective algorithms and solutions to the MRCPSP/max problem is still a lively field of research. One of the promising directions of such research is to take advantage of the parallel and distributed computation solutions, which are the feature of the contemporary multiple-agent systems.

The multiple-agent systems are an important and intensively expanding area of research and development. There is a number of multiple-agent approaches

proposed to solve different types of optimization problems. One of them is the concept of an asynchronous team (A-Team), originally introduced in [12]. The A-Team paradigm was used to develop the JADE-based environment for solving a variety of computationally hard optimization problems called E-JABAT [1]. E-JABAT is a middleware supporting the construction of the dedicated A-Team architectures based on the population-based approach. The mobile agents used in E-JABAT allow for decentralization of computations and use of multiple hardware platforms in parallel, resulting eventually in more effective use of the available resources and reduction of the computation time.

In this paper the E-JABAT-based A-Team architecture for solving the MRCPSP/max problem instances is proposed and experimentally validated. To solve instances of the MRCPSP/max different optimization agents are used. Optimization agents represent heuristic algorithms. The proposed architecture is an extension of the one proposed in [7]. The proposed approach has been validated experimentally.

Section 2 of the paper contains the MRCPSP/max problem formulation. Section 3 gives some information on E-JABAT environment. Section 4 provides details of the E-JABAT architecture implemented for solving the MRCPSP/max problem instances. Section 5 describes computational experiment carried-out with a view to validate the proposed approach. Section 6 contains conclusions and suggestions for future research.

2 Problem Formulation

In the multi-mode resource-constrained project scheduling problem with minimal and maximal time lags (MRCPSP/max) a set of $n + 2$ activities $V = \{0, 1, \dots, n, n + 1\}$ is considered. Each activity has to be processed without interruption to complete the project. The dummy activities 0 and $n + 1$ represent the beginning and the end of the project. For each activity $i \in V$ a set $M_i = \{1, \dots, |M_i|\}$ of (execution) modes is available. Activities 0 and $n + 1$ can be performed in only one mode: $M_0 = M_{n+1} = \{1\}$. Each activity $i \in V$ has to be performed in exactly one mode $m_i \in M_i$. The duration (processing time) of an activity i , $i \in V$ executed in m_i mode is denoted by d_{im_i} , $d_{im_i} \in Z_{\geq 0}$. The processing times of activity 0 and $n + 1$ equals 0, i.e. $d_{00} = d_{n+1,0} = 0$.

S_i and C_i stand for the start time and the completion time (of the performance) of activity i , respectively. If we define $S_0 = 0$, S_{n+1} stands for the project duration. Provided that activity i starts in mode m_i at time S_i , it is being executed at each point in time $t \in [S_i, S_i + d_{im_i})$.

Between the start time S_i of activity i , which is performed in mode $m_i \in M_i$, and the start time S_j of activity j ($i \neq j$), which is performed in mode $m_j \in M_j$, a minimum time lag $d_{im_i, jm_j}^{min} \in Z_{\geq 0}$ or a maximum time lag $d_{im_i, jm_j}^{max} \in Z_{\geq 0}$ can be given. Note, that a time lag between activity i and activity j depends on mode m_i as well as on mode m_j .

Activities and time lags are represented by an activity-on-node (AoN) network $N = \langle V, E, \delta \rangle$ with node set V , arc set E , and arc weight function δ . Each element

of node set V represents an activity. In the following, we do not distinguish between an activity and the corresponding node. An arc $\langle i, j \rangle \in E$ indicates that a time lag between S_i and S_j has to be observed. Arc weight function δ assigns to each arc $\langle i, j \rangle \in E$ a $|M_i| \times |M_j|$ matrix of arc weights as follow: for a minimum time lag d_{im_i, jm_j}^{min} we set $\delta_{im_i, jm_j} = d_{im_i, jm_j}^{min}$, and for a maximum time lag d_{im_i, jm_j}^{max} we set $\delta_{im_i, jm_j} = -d_{im_i, jm_j}^{max}$.

There are the set of renewable resources R^R and the set of nonrenewable resources R^N considered in this problem, $|R^R|, |R^N| \in Z_{>0}$. The availability of each renewable resource type $k \in R^R$ in each time period is R_k^R units. The availability of each nonrenewable resource type $k \in R^N$ is R_k^N units in total. Provided that activity i is performed in mode m_i , $r_{im_i, k}^R$ units of renewable resource $k \in R^R$ are used at each point in time at which activity i is being executed. Moreover, $r_{im_i, k}^N$ units of nonrenewable resource $k \in R^N$ are consumed in total. For activities 0 and $n + 1$ we set $r_{01k} = r_{n+11k} = 0$ for $k \in R^R$ and $r_{01k}^N = r_{n+10k}^N = 0$ for $k \in R^N$.

The solution of the problem is a schedule (M, S) consisting of the mode vector M and a vector of activities starting times $S = [S_0, \dots, S_{n+1}]$, where $S_0 = 0$ (project always begins at time zero). The mode vector assigns to each activity $i \in V$ exactly one mode $m_i \in M_i$ - execution mode of activity i . The start time vector S assigns to each activity $i \in V$ exactly one point in time as start time S_i where $S_0 = 0$ (project always begins at time zero). Precedence relations are described by the following formula: $S.S_j - S.S_i \geq \delta_{im_i, jm_j}$, where $\langle i, j \rangle \in E$.

The objective is to find a schedule (M, S) where precedence and resource constraints are satisfied, such that the schedule duration $T(S) = S_{n+1}$ is minimized. The detailed description of the problem can be found in [5] or [3]. The MRCPSP/max, as an extension of the RCPSP and RCPSP/max, belongs to the class of NP-hard optimization problems [2].

3 E-JABAT Environment

E-JABAT is a middleware allowing to design and implement A-Team architectures for solving various combinatorial optimization problems. The problem-solving paradigm on which the proposed system is based can be best defined as the population-based approach.

E-JABAT produces solutions to combinatorial optimization problems using a set of optimization agents, each representing an improvement algorithm. Each improvement (optimization) algorithm when supplied with a potential solution to the problem at hand, tries to improve this solution. An initial population of solutions (individuals) is generated or constructed. Individuals forming an initial population are, at the following computation stages, improved by independently acting agents. Main functionality of the proposed environment includes organizing and conducting the process of search for the best solution.

To perform the above described cycle two main classes of agents are used. The first class called OptiAgent is a basic class for all optimization agents. The second class called SolutionManager is used to create agents or classes of agents

responsible for maintenance and updating individuals in the common memory. All agents act in parallel. Each OptiAgent represents a single improvement algorithm (for example: local search, simulated annealing, tabu search, genetic algorithm etc.).

Other important classes in E-JABAT include: Task representing an instance or a set of instances of the problem and Solution representing the solution. To initialize the agents and maintain the system the TaskManager and Platform-Manager classes are used. Objects of the above classes also act as agents. Up to now the E-JABAT environment has been used to solve instances of the following problems: the resource-constrained project scheduling problem (RCPSP), the traveling salesman problem (TSP), the clustering problem (CP), the vehicle routing problem (VRP).

E-JABAT environment has been designed and implemented using JADE (Java Agent Development Framework), which is a software framework supporting the implementation of multi-agent systems. More detailed information about E-JABAT environment and its implementations can be found in [6] and [1].

4 E-JABAT Architecture for Solving the MRCPSP/max Problem

E-JABAT environment was successfully used for solving the RCPSP, MRCPSP and RCPSP/max problems [7]. In the proposed approach the new agents, classes describing the problem and ontologies have been implemented for solving the discussed problem. The above forms the package called JABAT.MRCPSPmax.

Classes describing the problem are responsible for reading and preprocessing the data and generating random instances of the problem. The discussed set includes the following classes:

- MRCPSPmaxTask inheriting from the Task class and representing the instance of the problem,
- MRCPSPmaxSolution inheriting from the Solution class and representing the solution of the problem instance,
- Activity representing the activity of the problem,
- Mode representing the activity mode,
- Resource representing the renewable or nonrenewable resource,
- PredSuccA representing the predecessor or successor of the activity.
- PredSuccM and PredSuccT representing the matrix of arc weights. The matrix is needed to describe time lags for all pairs of modes of each two activities connected by the arc in the network N .

The next set includes classes allowing for definition of the vocabulary and semantics for the content of messages exchange between agents. In the proposed approach the messages include all data representing the task and solution. The discussed set includes the following classes:

- MRCPSPmaxTaskOntology inheriting from the TaskOntology class,
- MRCPSPmaxSolutionOntology inheriting from the SolutionOntology class,

The last set includes classes describing the optimization agents. Each of them includes the implementation of an optimization heuristic used to solve the problem. All of them are inheriting from OptiAgent class. In the considered case this set includes the following classes:

- optiLSA denoting the Local Search Algorithm (LSA),
- optiPRA denoting Path Relinking Algorithm (PRA),
- optiCA denoting Crossover Algorithm (CA),

The algorithms proposed in [7] has been rebuild and adapted to solving the MRCPSP/max problem. The main modification is introduction of the mode handling mechanism. In all the objective is to find best feasible schedule (M, S) . The procedure of finding a new solution from the schedule (M, S) is based on the SGSU (Serial Generation Scheme with Unscheduling) described in [8] with several different priority rules. The pseudo-codes of the algorithms are shown in Figures 1, 2, 3 respectively.

```

LSA(initialSchedule, startActivityPosition, iterationNumber)
{
  S = initialSchedule //i.e.  $(M, S)$  with ordered activity list
  pi = startActivityPosition //thus i is the start activity
  it = iterationNumber
  if( $S.S_{n+1} > 0$ ) minS =  $S.S_{n+1}$ ; else minS =  $\infty$ 
  while(it>0 || not all  $i \in V$  are checked)
  {
    minSit = minS
    pi =  $++pi \% (n - 2) + 1$ 
    for(pj = startActivityPosition; pj <  $n + 1$ ; pj += 3)
      for(all  $k \in S.M_i$ )
      {
        makeMove(S, pi, pj, k)
        searchSolutionBySGSU(S)
        if( $S.S_{n+1} \geq 0$  &&  $S.S_{n+1} \leq minSit$ )
          { minSit =  $S.S_{n+1}$ ; bestSit = S }
        reverseMove(S, pi, pj, k)
      }
    if(minSi > 0 && minSi < minS)
    {
      minS = minSit
      bestS = bestSit
    }
    else it-
  }
}

```

Fig. 1. Pseudo-code of the LSA algorithm

```

PRA(initialSchedule1, initialSchedule2)
{
  S1 = initialSchedule1
  S2 = initialSchedule2
  S = S1
  if(S1.Sn+1 > 0 && S1.Sn+1 < bestS.Sn+1) bestS = S1
  if(S2.Sn+1 > 0 && S2.Sn+1 < bestS.Sn+1) bestS = S2
  for(S.pi = 1; S.pi < n + 1; S.pi++)
  {
    S.pi = position of S.i in S2
    S.pj = position of S2.i in S
    makeMove(S, S.pj, S.pi, S.mi)
    searchSolutionBySGSU(S)
    if(S.Sn+1 > 0 && S.Sn+1 < bestS.Sn+1) bestS = S
  }
}

```

Fig. 2. Pseudo-code of the PRA algorithm

```

CA(initialSolution1, initialSolution2, step)
{
  if(S1.Sn+1 > 0) bestS = S1
  if(S2.Sn+1 > 0 && S2.Sn+1 < S1.Sn+1) bestS = S2
  for(cp = 2; cp < n - 2; cp += step)
  {
    makeCrossover(S, S1, S2, cp)
    searchSolutionBySGSU(S)
    if(S.Sn+1 > 0 && S.Sn+1 < bestS.Sn+1) bestS = S
  }
}

```

Fig. 3. Pseudo-code of the CA algorithm

In the pseudo-codes S denotes the schedule (M, S) with ordered list of activities. The condition $S.S_{n+1} > 0$ means that S assures a feasible solution. All presented algorithms can use initial schedules which does not necessarily guarantee obtaining feasible solutions using `searchSolutionBySGSU` procedure.

The LSA (Figure II) is a simple local search algorithm which finds local optimum by moving each activity with each possible mode to all possible places in the schedule. For each combination of activities the value of possible solution is calculated. The best schedule is returned. The parameter *iterationNumber* means a maximum number of iterations in which no improvement is found. Two procedures are used to make the moves. The `makeMove(S, pi, pj, nmi)` means moving the activity form position pi in the schedule (activity list) S to position pj and simultaneously changing the chosen activity mode m_i to nm_i . The `reverseMove(S, pi, pj, nmi)` means cancelling the move i.e. moving the activity

from position p_j to position p_i and simultaneously changing the activity mode from nm_i to previous one m_i .

The PRA (Figure 2) is an implementation of the path-relinking algorithm. For a pair of solutions a path between them is constructed. The path consists of schedules obtained by carrying out a single move from the preceding schedule. The move is understood as moving one of the activities to a new position simultaneously changing its mode. For each schedule in the path the value of the respective solution is checked. The best schedule is remembered and finally returned. In the PRA the `makeMove` procedure is used described above.

The CA (Figure 3) is an algorithm based on the idea of one point crossover operator. For a pair of solutions one point crossover is applied. The *step* argument determines the frequency the operation is performed. The `makeCrossover($S, S1, S2, cp$)` constructs the S schedule using one point crossover operator to two initial schedules $S1$ and $S2$ with crossover point cp .

All optimization agents co-operate together using the E-JABAT common memory. The initial population in the common memory is generated randomly with the exception of a few individuals which are generated by heuristics based on the priority rules 8 and `searchSolutionBySGSU` procedure. Because it might be difficult to obtain feasible solution for some RCPSP/max problem instances, the random drawing of an individual could be repeated several times. If this does not produce enough feasible solutions the infeasible ones are added to the population in the common memory. In some instances the initial population consist only of the infeasible solutions. Individuals in the common memory are represented as schedules of activities. The final solution is obtained from the schedule by `searchSolutionBySGSU`.

The time and frequency an agent of each kind receives a solution or set of solutions from the common memory with a view to improve its quality is determined by the strategy. For solving the MRCPSP/max problem instances the strategy where individuals forwarded to optimization agents for improvement are randomly chosen from the population stored in the common memory has been used. Such individuals are sent to optimization agents ready to start searching for a better solution. After computation the improved individual replaces the worst one stored in the common memory.

5 Computational Experiment Results

To validate the proposed approach and to evaluate the effectiveness of the optimization agents the computational experiment has been carried out using benchmark instances of MRCPSP/max from PSPLIB 10 - test set mm100 with activities carried out in 3, 4 and 5 modes. The set includes 270 problem instances. The experiment involved computation with the fixed number of optimization agents representing LSA, PRA and CA algorithms, fixed population size, and the limited time period allowed for computation.

The discussed results have been obtained using 3 optimization agents: one of each kind. Population of solutions in the common memory consisted of 100 individuals. The computation has been stopped after 5 minutes. The optimization

Table 1. Performance characteristics of the proposed A-Team for benchmark test set mm100

#Modes	Mean RE	% FS	Mean total CT [s]	Mean CT [s]
3	66.2%	86%	304.6	99.2
4	85.6%	88%	305.5	95.4
5	104.2%	92%	304.9	99.0

Table 2. Literature reported results – TS_{DR} [4]

#Modes	Mean RE	% FS	Mean CT [s]
3	40%	53%	100
4	91%	61%	100
5	164%	67%	100

Table 3. Literature reported results – Prio [4]

#Modes	Mean RE	% FS	Mean CT [s]
3	63%	100%	100
4	113%	100%	100
5	170%	100%	100

Table 4. Percentage of the improved and best improved solutions obtained by the agents in the proposed approach

#Modes	LSA	PRA	CA	Any	No one
Percents of improved solutions					
3	39%	46%	48%	51%	49%
4	47%	57%	46%	64%	36%
5	64%	60%	53%	77%	23%
Percents of best improved solutions					
3	19%	20%	16%	32%	68%
4	17%	24%	10%	38%	62%
5	21%	20%	8%	40%	60%

algorithms have had a fixed parameter values but some of them were chosen randomly. In the case of LSA the *startActivityPosition* was chosen randomly from the interval $[1, n]$, for the *iterationNumber* the number was chosen randomly from the interval $[3, 5]$. In the case of CA the *step* is set to 3.

The following characteristics have been calculated and recorded: mean relative error (Mean RE) calculated as the deviation from the lower bound, percent of feasible solutions (% FS), mean total computation time (Mean total CT) and mean computation time required to find the best solution (Mean CT). Each

instance has been solved five times and the results have been averaged over these solutions. The results are presented in Table 1. These results are compared with the results reported in the literature (Tables 2, 3) obtained by tabu search algorithm TS_{DR} and heuristic based on multi-pass priority-rule method with backplanning Prio proposed in 4.

Additionally, the influence of each agent on the results was evaluated as the average percent of individuals which were improved by it and the average percent of the current best solutions found by it. The results in terms of the percent of solutions improved by each agent, percent of solutions improved by any agent and percent of non-improved solutions are presented in Table 4.

Experiment has been carried out using nodes of the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core 1.4 GHz with 12 MB L3 cache processors and with Mellanox InfiniBand interconnections with 10Gb/s bandwidth. During the computation one node per agent was used.

6 Conclusions

Experiment results show that the proposed E-JABAT based A-Team implementation is an effective tool for solving instances of the MRCPSP/max problem. Presented results are comparable with solutions known from the literature. It can be also noticed that they are obtained in a comparable time. Time comparisons are however difficult since the discussed implementation have been using different numbers and kinds of processors. In case of the agent-based environments the significant part of the time is used for agent communication which has an influence on both - computation time and quality of the results.

Evaluating particular agents it can be noted that their contributions towards improving the current best solution have been similar, the differences in this respect for the same number of nodes are about 10%. Each agent has been producing the improved solutions for 39% to 64% of solutions from the population. Considering the best solutions found, the differences are small. Each agent has found the best solutions for 8% to 24% of solutions from the population.

Considering effectiveness of agents, the most effective one is the PRA one proposed in this paper. However the experiment results show that the other two agents LSA and CA could find the best solutions for different individuals than PRA, so co-operation of agents becomes effective.

Future research will concentrate on implementing more and more sophisticated optimization agents, as well as finding the best configuration of the heterogeneous agents. Since the E-JABAT gives a possibility to run more than one copy of each agent it is interesting which agents should or should not be replicated to improve the results. Additionally, testing and adding to E-JABAT more different optimization agents and improving the existing ones could be considered.

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education grant no. N N519 576438 for years 2010–2013. Calculations have been performed in the Academic Computer Centre TASK in Gdansk.

References

1. Barbucha, D., Czarnowski, I., Jedrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: E-JABAT An Implementation of the Web-Based A-Team. In: Nguyen, N.T., Jain(eds, L.C. (eds.) *Intelligent Agents in the Evolution of Web and Applications*, pp. 57–86. Springer, Heidelberg (2009)
2. Bartusch, M., Mohring, R.H., Radermacher, F.J.: *Scheduling Project Networks with Resource Constraints and Time Windows*. *Annual Operational Research* 16, 201–240 (1988)
3. Barrios, A., Ballestin, F., Valls, V.: A double genetic algorithm for the MR-CPSP/max. *Computers and Operations Research* 38, 33–43 (2011)
4. Heilmann, R.: Resource-Constrained Project Scheduling: a Heuristic for the Multi-Mode Case. *OR Spektrum* 23, 335–357 (2001)
5. Heilmann, R.: A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research* 144, 348–365 (2003)
6. Jedrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) *ICCS 2006*. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
7. Jedrzejowicz, P., Ratajczak-Ropel, E.: Solving the RCPSP/max Problem by the Team of Agents. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009*. LNCS, vol. 5559, pp. 734–743. Springer, Heidelberg (2009)
8. Neumann, K., Schwindt, C., Zimmermann, J.: *Project Scheduling with Time Windows and Scarce Resources*, 2nd edn. Springer, Heidelberg (2003)
9. Nonobe, K., Ibaraki, T.: A Tabu Search Algorithm for a Generalized Resource Constrained Project Scheduling Problem. In: *MIC 2003: The fifth metaheuristics International Conference*, pp. 551–556 (2003)
10. PSPLIB, <http://129.187.106.231/psplib>
11. Reyck, B., Herroelen, W.: The Multi-Mode Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. *European Journal of Operational Research* 119, 538–556 (1999)
12. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: *Asynchronous Teams: Cooperation Schemes for Autonomous, Computer-Based Agents*. Technical Report EDRC 18-59-96. Carnegie Mellon University, Pittsburgh (1996)

An Agent-Based Guided Local Search for the Capacited Vehicle Routing Problem

Dariusz Barbucha

Department of Information Systems
Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
barbucha@am.gdynia.pl

Abstract. The main contribution of the paper is to propose and validate a new agent-based method of solving instances of the Capacited Vehicle Routing Problem (CVRP). The approach adopts a guided local search metaheuristic and combines it with an asynchronous team (A-Team) concept. A-Team assumes that a collection of software agents, each representing a particular problem solving method, cooperate to solve a problem by dynamically evolving a population of solutions. In suggested implementation each software agent carries out a guided local search. The paper contains the CVRP formulation, an overview of the dedicated multi-agent framework and a description of the proposed implementation for solving CVRP. The approach is validated experimentally and results of computational experiment are included in the final part of the paper.

Keywords: capacited vehicle routing problem, guided local search, multi-agent systems, asynchronous team.

1 Introduction

Capacited Vehicle Routing Problem (CVRP) is probably the best known optimization problem from transportation and logistics area, in which a set of given customers is to be served by the fleet of vehicles in order to minimize the service cost and satisfying several customer's and vehicle's constraints. It may be defined as an undirected graph $G = (V, E)$, where $V = \{0, 1, \dots, N\}$ is the set of nodes and E is a set of edges. Node 0 is a central depot with NV identical vehicles of capacity W . Each other node $i \in V - \{0\}$ denotes customer with a non-negative demand d_i . Each link $(i, j) \in E$ denotes the shortest path from customer i to j and is described by the cost c_{ij} of travel from i to j by shortest path ($i, j = 1 \dots, N$). It is assumed that $c_{ij} = c_{ji}$. The goal is to find vehicle routes which minimize total cost of travel (or travel distance) and satisfy following constraints:

- each route starts and ends at the depot,
- each customer is serviced exactly once by a single vehicle,
- the total load on any vehicle associated with a given route does not exceed vehicle capacity.

There have been important advances in the development of exact and approximate algorithms for solving CVRP. Because of the fact that the problem belongs to the NP-hard class, most of them are of heuristic nature. The general classification of existing heuristics distinguishes a group of *classical approaches* (construction or improvement ones) and *metaheuristics*.

Classical heuristics, like Clarke's and Wright's *savings* algorithm [6], Gillett's and Miller's *sweep* algorithm [9], Fisher's and Jaikumar's *cluster-first route-second* algorithms [7], or various *multi-route edge exchange* algorithms of Thompson and Psaraftis [22], van Breedam [24], and Kinderwater and Savelsbergh [13] are often simple and fast but the quality of solutions obtained by them is as a rule not satisfactory.

On the other hand, metaheuristics provide much better solutions, especially in case of large-scale instances, although they require much more computational resources and have to be fine-tuned in order to fit a particular problem. The main group of metaheuristics which have been proposed with a success during the last twenty years for solving CVRP are *tabu search* approaches. These include implementations provided by Gendreau et al. [8], Taillard [19], Xu and Kelly [27], Rochat and Taillard [18], and Toth and Vigo [23]. Other classical metaheuristics, like evolutionary algorithms, simulated annealing, ant colony algorithms, or guided local search, sometimes combined with other local search techniques, have been also the milestones in effective solving of instances of CVRP. Implementations of Mester and Bräysy [15], Prins [16], Berger and Berkaoui [4], and Reinmann et al. [17] are worth mentioning in this respect.

Recent surveys of exact and heuristic approaches for solving classical CVRP are presented for example by Laporte et al. [14], and Golden et al. [10].

During last years, a growing number of decentralized approaches composed of multiple autonomous components (*multi-agent systems* - MAS) [11] dedicated for solving various transportation problems have arisen. Some of them refer directly to the solving one of the problem from VRP class. Such approaches are presented, for example by Thangiah et al. [21] and Barbucha [3], where different agent-based architectures are proposed for solving classical VRP, and, for example, by Barbucha [2], who propose its agent-based solution for dynamic version of Vehicle Routing Problem. Results obtained by various approaches based on multi-agent paradigm suggest that this area of research on effective solving of instances of Vehicle Routing Problem seems to be promising.

The main contribution of the paper is to propose a new agent-based approach for solving Capacitated Vehicle Routing Problem which combines various *guided local search* metaheuristics within a one *multi-agent* framework based on the concept of an *asynchronous team* (A-Team). The rest of the paper is organized as follows. Section 2 contains a description of agent-based guided local search framework. Details of the proposed approach based on the suggested framework for solving VRP are given in Section 3. Section 4 shows the plan of the computational experiment which has been carried out in order to validate an effectiveness of the proposed approach and reports on the obtained results. And finally, conclusions included in Section 5 end the paper.

2 Agent-Based Guided Local Search Framework

2.1 Overview

The proposed agent-based approach for solving CVRP is based on asynchronous team (A-Team) concept, where a collection of software agents cooperates to solve an instance of the problem by dynamically evolving the population of solutions stored in the common memory [20]. Agents cooperate by sharing access to populations of candidate solutions. Agents work to create, modify or remove solutions from the population. The quality of the solutions gradually evolves over time as improved solutions are added and poor solutions are removed. Within an A-Team, agents are autonomous and asynchronous. Each agent encapsulates a particular problem-solving method along with the methods to decide when to work, what to work on and how often to work. In general, an implemented problem solving method may be inspired by natural phenomena including, for example, evolutionary processes, as well as local search techniques.

Main functionality of the proposed approach is organizing and conducting the process of search for the best solution. It is organized as a sequence of steps, including initialization and improvement phases. At first the initial population of solutions is generated. Individuals forming the initial population are, at the following computation stages, improved by independently acting autonomous agents, thus increasing chances for reaching the global optimum. Finally, when the stopping criterion is met, the best solution from the population is taken as the result.

The strength of A-Team applications for solving particular optimization problem stems from a problem-solving method encapsulated in each optimization agent. In the presented A-Team implementation it has been decided to include Guided Local Search metaheuristics [25] with various local search procedures implemented as improvement methods.

The pseudo-code of the suggested approach is shown as the Algorithm 1.

Algorithm 1. Agent-Based Guided Local Search Procedure (ABGLS)

- 1: Generate an initial population of solutions (individuals) and store them in the common memory
 - 2: **while** (stopping criterion is not met) **do** {in parallel}
 - 3: Select individual from the common memory
 - 4: Select an optimization agent carrying-out guided local search improvement algorithm and execute it improving the selected individual
 - 5: After improvement, if new individual is accepted, store it back in the common memory
 - 6: **end while**
 - 7: Take the best solution from the population as the result
-

2.2 Guided Local Search

Guided Local Search (GLS) originally proposed by Voudouris and Tsang [25] as a general optimization technique suitable for a wide range of computationally

hard optimization problems turned out as one of the most effective methods for solving difficult problems such as, for example, Traveling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP) [26]. Also implementation of GLS for the Vehicle Routing Problem was proposed, for example, in [1], [12] but without a multi-agent context.

Guided Local Search focuses on the exploitation problem and uses search-related information to effectively guide local search heuristics in the vast search spaces. This is achieved by augmenting the objective function of the problem to be minimized with a set of penalty terms which are dynamically manipulated during the search process to control the heuristic to be guided [26].

GLS employs solution features to characterize solutions. A *solution feature* - f_i ($i = 1, \dots, M$, and M is the number of features defined over solutions) can be any solution property that satisfies the simple constraint that is a non-trivial one. Constraints on features are introduced or strengthened on the basis of information about the problem and also the course of the local search.

Each feature f_i is represented for each solution $s \in S$, where S - search space, by an *indicator function* - I_i in the following way:

$$I_i(s) = \begin{cases} 1 & \text{if solution } s \in S \text{ has property } i, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The *cost of features* - c_i ($i = 1, \dots, M$) is an information pertaining to the problem. It represents the direct or indirect impact of the corresponding solution properties on the solution cost.

GLS augments the original cost function of the problem g to include a *set of penalty terms* - p_i ($i = 1, \dots, M$) and passes this, instead of the original one, for minimization by the local search procedure. The newly formed cost function h is called the *augmented cost function* and it is defined as follows:

$$h(s) = g(s) + \lambda \sum_{i=1}^M p_i * I_i(s), \quad (2)$$

where λ is a parameter for controlling the strength of constraints with respect to the actual solution cost. It represents the relative importance of penalties with respect to the solution cost and it provides means to control the influence of the information on the search process. The penalty parameter p_i gives the degree up to which the solution feature f_i is constrained.

Local search performed with the augmented cost function focuses its attention on promising regions of the search space. Iterative calls are made to carry out the local search. Each time local search gets caught in a local minimum, the penalties are modified and local search is called again to minimize the modified cost function.

The pseudocode of Guided Local Search procedure is presented as the Algorithm 2.

Algorithm 2. Guided Local Search (GLS)

```

1:  $k = 0$ ;
2:  $s_k \leftarrow \text{InitSolution}()$ ;
3:  $s^* \leftarrow s_k$ 
4: for  $i = 1$  to  $M$  do
5:    $p_i \leftarrow 0$ ;
6: end for
7: while (stopping criterion is not met) do
8:    $h \leftarrow g + \lambda \sum_{i=1}^M p_i * I_i(s_k)$ ;
9:    $s_{k+1} \leftarrow \text{LocalSearch}(s_k, h)$ ;
10:  for  $i = 1$  to  $M$  do
11:    let  $c_i$  be a cost of feature  $i$ 
12:    let  $util_i \leftarrow I_i(s_{k+1}) * \frac{c_i}{1+p_i}$ ;
13:  end for
14:  for all  $i$  such that  $util_i$  is maximum do
15:     $p_i \leftarrow p_i + 1$ ;
16:  end for
17:  if ( $g(s_{k+1}) < g(s^*)$ ) then
18:     $s^* \leftarrow s_{k+1}$ 
19:  end if
20:   $k \leftarrow k + 1$ ;
21: end while
22:  $s^* \leftarrow \text{LocalSearch}(s^*, g)$  // best solution found with respect to cost function  $g$ 

```

3 An Agent-Based Implementation of the Guided Local Search for the Capacited Vehicle Routing Problem

According to a general scheme of the A-Team, the presented agent-based implementation of GLS for CVRP requires definition of several elements such as: *representation of an individual*, methods of *creating an initial population of individuals*, methods of *managing population of individuals* including: selection, acceptance and updating, *stopping criterion*, and form of *optimization agents*.

3.1 Representation of an Individual

A permutation of N numbers (representing customers), where the order of numbers reflects the order in which customers are visited, has been chosen as an individual's representation (path representation of TSP).

3.2 Initial Population of Individuals

The proposed approach results in creation of the initial population which is based on polar representation of each vertex (here customer) and uses an idea originated from *split* phase of the *sweep* algorithm of Gillett and Miller [9].

First, each vertex $i \in V$ is transformed from cartesian coordinates to polar coordinates (θ_i, ρ_i) , where θ_i is the angle and ρ_i is the ray length.

Generation of each individual in the population starts from randomly choosing an arbitrary vertex i^* and assigning a value $\theta_{i^*} = 0$ to it. Next, the remaining angles centered at 0 from the initial ray $(0, i^*)$ are computed and the vertices are ranked in increasing order of their θ_i value. Resulting ranking determines an individual.

The process of assignment vertices to clusters (routes) starts from the first unrouted vertex having the smallest angle, and repeatedly assigns vertices to the cluster (route) as long as its capacity or the maximal route length is not exceeded. The process is repeated until end of the individual is reached.

3.3 Management of the Population of Individuals

For the purpose of the research the following method of *selecting* a solution from the population has been implemented. The solution manager chooses from population a random solution and next sends it to the optimization agent(s). Additionally, in order to prevent the same solution from being sent to the other optimization agent in the asynchronous mode, it is blocked for a period of time.

In the suggested approach, an *updating* mechanism gives the solution manager opportunity to replace the worst solution from current population by a solution currently received from optimization agent. Additionally, if last consecutive five solutions received from the optimization agents did not improve existing solutions in population, the worst solution is removed from the population and a newly generated one is added to the pool of individuals. Only solutions locally or globally improved received from optimization agents are *accepted* and added to the common memory.

3.4 Stopping Criterion

In the proposed implementation of cooperative search the system stops after a given number of seconds.

3.5 Guided Local Search Optimization Agents

As it was mentioned, each optimization agent included in the proposed multi-agent framework shares a general schema of GLS. Such schema requires definitions of feature set, feature costing, penalty factor and local search procedures.

Similar to [26] and [12], the arcs $(i, j) \in E$ were chosen as a feature to penalize and, hence, the cost of feature is equal to c_{ij} ($i, j = 0, \dots, N$). Penalty factor λ was set to 0.15. Three guided local search optimizations agents each using a different local search (*LS*) method have been implemented:

- $LS_1()$ - an implementation of the *3-opt* procedure which operates on a single route. First, for a given feasible solution (tour) three edges are removed and next remaining edges are reconnected in all possible ways until a new feasible solution (tour) is obtained.
- $LS_2()$ - an implementation of the dedicated *local search method* based on the interchange or move of randomly selected customers between two selected

routes. Let (v, u) , where $u, v \in \{0, 1, 2\}$ be an ordered pair denoting the number of customers to interchange or move between two selected routes. For example, $(2, 0)$ indicates a shift of two customers from the first route to the second route, while $(2, 2)$ indicates an exchange of two customers between routes. According to the given pair randomly selected v and u customers from first and second route, respectively, are shifted or exchanged between routes.

- $LS_3()$ - another implementation of the dedicated *local search method* which operates also on two randomly selected routes, but opposite to the previous method, here customers are selected not randomly but in accordance to their distance to the centroid of their original route. Similar to $LS_2()$, let (v, u) , where $u, v \in \{0, 1, 2\}$ be an ordered pair denoting the number of customers to interchange or move between two selected routes. First, v (u , respectively) customers from first (second) route for which the distance between them and the centroid of their route is the greatest are removed from their original routes. Next, they are moved to the opposite routes and inserted in them on positions, which give the smallest distance between newly inserted customer and the centroid of this route.

4 Computational Experiment

Computational experiment aimed at validating effectiveness of the proposed agent-based guided local search approach and evaluating how the guided search influences computation results, measured as mean relative error (MRE) from the optimal (or the best one) solution.

The experiment has been divided into two parts. In the first part, all optimization agents (with and without the guided local search implemented) have been running separately. Optimization agents have been named as $OA_LS_i()$ and $OA_GLS_i()$, with 3 copies of each kind ($i = 1, 2, 3$). Next, in the second part of the experiment, all optimization agents (with or without implementation of the guided local search) have been running collectively, forming a team. In both parts, results obtained for cases with guided local search implemented in each optimization agent were compared with the results for cases without implementation of guided local search.

The above cases were tested on 7 instances of Christofides et al. [5] (named $vrpnc1$ - $vrpnc5$ and $vrpnc11$ - $vrpnc12$), each of them containing 50-199 customers with only capacity restrictions. Each of the 7 instances was repeatedly solved 10 times using the proposed Agent-Based Guided Local Search algorithm (ABGLS) and the mean results from these runs were recorded.

All computations have been carried out on the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core with 12 MB L3 cache processors with Mellanox InfiniBand interconnections with 10Gb/s bandwidth.

All runs have been stopped after the limit of 5 minutes has been reached.

The experiment results are summarized in Tables 1 and 2. Tab. 1 shows mean relative errors from the best known solution obtained by proposed approach in

Table 1. The performance of the proposed agent-based approach (MRE) calculated separately for each optimization agent with and without implemented guided search, respectively

AGENTS	INSTANCE							AVG
	<i>vrpnc1</i> (50)	<i>vrpnc2</i> (75)	<i>vrpnc3</i> (100)	<i>vrpnc4</i> (150)	<i>vrpnc5</i> (199)	<i>vrpnc11</i> (100)	<i>vrpnc12</i> (120)	
<i>OALS</i> ₁ ()	1.14%	3.63%	4.03%	4.58%	6.45%	2.38%	2.93%	3.59%
<i>OALS</i> ₂ ()	0.03%	0.91%	1.62%	3.98%	4.12%	1.19%	2.12%	2.00%
<i>OALS</i> ₃ ()	0.02%	0.80%	2.41%	2.98%	3.72%	1.77%	2.06%	1.97%
<i>OAGLS</i> ₁ ()	0.41%	2.43%	3.54%	4.43%	5.68%	2.13%	2.45%	3.16%
<i>OAGLS</i> ₂ ()	0.00%	0.82%	1.15%	2.88%	3.98%	0.94%	1.43%	1.71%
<i>OAGLS</i> ₃ ()	0.00%	0.81%	2.21%	2.31%	3.70%	1.92%	2.32%	1.89%

Table 2. The performance of the proposed agent-based approach (MRE) calculated for two teams of optimization agents with and without implemented guided search, respectively

AGENTS	INSTANCE							AVG
	<i>vrpnc1</i> (50)	<i>vrpnc2</i> (75)	<i>vrpnc3</i> (100)	<i>vrpnc4</i> (150)	<i>vrpnc5</i> (199)	<i>vrpnc11</i> (100)	<i>vrpnc12</i> (120)	
<i>OALS</i> ₁ ()								
<i>OALS</i> ₂ ()	0.00%	0.72%	1.33%	2.53%	3.72%	1.02%	1.94%	1.61%
<i>OALS</i> ₃ ()								
<i>OAGLS</i> ₁ ()								
<i>OAGLS</i> ₂ ()	0.00%	0.65%	1.07%	2.13%	2.19%	1.14%	1.38%	1.22%
<i>OAGLS</i> ₃ ()								

the first part of experiment. Here, mean relative errors are calculated for each instance and separately for each optimization agent with and without implemented guided search, respectively. In turn, Tab. 2 shows mean relative errors from the best known solutions obtained by proposed approach in the second part of the experiment, where two teams of optimization agents (with and without implemented guided search, respectively) have solved instances of the problem. In both parts of experiment, presented results are averaged over all tested instances.

Analysis of the results presented in both tables allows one to conclude that results produced by proposed multi-agent approach for solving CVRP are quite good with mean relative error between 0% and 4%, strongly depending on size of the problem and kind of local search method used in each optimization agent.

By comparing the results obtained by team of all three optimization agents solving in the same time the instance of the problem with the results of each optimization agent working separately, the superiority of the first one has been observed.

And finally, it has been also observed that incorporation of guided local search into optimization agents solving the problems has brought a lot of benefits in

term of MRE. It turned out that in almost all cases with guided search the observed results were better than in case without guided search. Moreover, the above behavior has been observed not only for team of agents but also where all agents have been running separately.

5 Conclusions

A new agent-based method of solving instances of the Capacitated Vehicle Routing Problem has been proposed in the paper. The presented approach adopts a guided local search metaheuristic and combines it with an asynchronous team concept. The computational experiment confirmed effectiveness of the proposed method. Also, the influence of guided local search on behavior of optimization agents has been observed. The mean relative errors from the optimal (or the best one) solution are less than for cases without guided search included in optimization agents for almost all considered instances.

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education grant no. N N519 576438. Calculations have been performed in the Academic Computer Centre TASK in Gdansk.

References

1. Backer, B.D., Furnon, V., Prosser, P., Kilby, P., Shaw, P.: Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics* 6(4), 501–523 (2000)
2. Barbucha, D., Jędrzejowicz, P.: Agent-Based Approach to the Dynamic Vehicle Routing Problem. In: Demazeau, Y., Pavon, J., Corchado, J.M., Bajo, J. (eds.) 7th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems, *Advances in Intelligent and Soft Computing*, vol. 55, pp. 169–178. Springer, Berlin (2009)
3. Barbucha, D.: Cooperative Solution to the Vehicle Routing Problem. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS, vol. 6071, pp. 180–189. Springer, Heidelberg (2010)
4. Berger, J., Barkaoui, M.: A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society* 54, 1254–1262 (2003)
5. Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.): *Combinatorial optimization*. John Wiley, Chichester (1979)
6. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, 568–581 (1964)
7. Fisher, M.L., Jaikumar, R.: A generalized assignment heuristic for vehicle routing. *Networks* 11, 109–124 (1981)
8. Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem. *Management Science* 40, 1276–1290 (1994)
9. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 22, 240–349 (1974)
10. Golden, B.L., Raghavan, S., Wasil, E.A. (eds.): *The Vehicle Routing Problem: Latest Advances and New Challenges*. *Operations Research Computer Science Interfaces Series*, vol. 43. Springer, Heidelberg (2008)

11. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems* 1, 7–38 (1998)
12. Kilby, P., Prosser, P., Shaw, P.: Guided local search for the vehicle routing problem. In: Voss, S., Martello, S., Osman, I.H., Roucairol, C. (eds.) *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pp. 473–486. Kluwer Academic Publishers, Dordrecht (1999)
13. Kinderwater, G.A.P., Savelsbergh, M.W.P.: Vehicle routing: handling edge exchanges. In: Aarts, E.H.L., Lenstra, J.K. (eds.) *Local Search in Combinatorial Optimization*, pp. 337–360. Wiley, Chichester (1997)
14. Laporte, G., Gendreau, M., Potvin, J., Semet, F.: Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7, 285–300 (2000)
15. Mester, D., Bräysy, O.: Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers and Operations Research* 34, 2964–2975 (2007)
16. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research* 31, 1985–2002 (2004)
17. Reimann, M., Doerner, K., Hartl, R.F.: D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers and Operations Research* 31, 563–591 (2004)
18. Rochat, Y., Taillard, E.D.: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics* 1(1), 147–167 (1995)
19. Taillard, E.D.: Parallel iterative search methods for vehicle routing problems. *Networks* 23, 661–673 (1993)
20. Talukdar, S., Baeretzen, L., Gove, A., de Souza, P.: Asynchronous teams: Cooperation schemes for autonomous agents. *Journal of Heuristics* 4, 295–321 (1998)
21. Thangiah, S.R., Shmygelska, O., Mennell, W.: An agent architecture for vehicle routing problem. In: *Proc. of the ACM Symposium on Applied Computing (SAC 2001)*, Las Vegas, pp. 517–521 (2001)
22. Thompson, P.M., Psaraftis, H.N.: Cyclic transfer algorithms for the multivehicle routing and scheduling problems. *Operations Research* 41, 935–946 (1993)
23. Toth, P., Vigo, D.: The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing* 15(4), 333–346 (2003)
24. Van Breedam, A.: An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints. Ph.D. dissertation, University of Antwerp (1994)
25. Voudouris, C., Tsang, E.: Partial Constraint Satisfaction Problems and Guided Local Search. In: *Proc. of 2nd Int. Conf. on Practical Application of Constraint Technology (PACT 1996)*, London, pp. 337–356 (1996)
26. Voudouris, C., Tsang, E.: Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research* 113, 469–499 (1999)
27. Xu, J., Kelly, J.P.: A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30, 379–393 (1996)

An Agent-Based Framework for Solving an Equity Location Problem

Maria Barbati¹, Giuseppe Bruno¹, and Andrea Genovese²

¹ University of Naples Federico II, Department of Business and Management Engineering (DIEG), Piazzale Tecchio 80, 80125 - Naples Italy
{`maria.barbati, giuseppe.bruno`}@unina.it

² University of Sheffield, Management School, Mappin Street 9, S1 4DT - Sheffield UK
`a.genovese@shef.ac.uk`

Abstract. Agent-Based Models have been recently proposed for solving optimization problems as an alternative to classical optimization methodologies or in combination with them; in particular this approach is useful when the considered problem is defined on a domain that is, to some extent, distributed, complex and heterogeneous.

In this paper, we present an adaptation of a general Agent-Based framework for modeling and solving a location problem where the objective function is a measure of “equity” from the demand points to the set of facilities (the Gini Coefficient). We present some computational results confirming the suitability and the effectiveness of the proposed approach.

Keywords. Equity Location Problems, Agent-Based models, Meta-Heuristics.

1 Introduction

An agent is a computational system interacting with an environment in order to achieve own objectives [1]. Agent-Based Models (ABMs) consist of a set of agents characterized by attributes which interact with each other through the definition of appropriate rules in a given environment. The behavior of an agent belonging to an ABM (see [2]) is characterized by independence (i.e. each agent acts without the direct control of other devices), social ability (i.e. agents interact each other through a communication language), re-activeness (i.e. agents react to signals coming from the environment) and pro-activeness (i.e. agents take the initiative in order to satisfy their design objectives).

For these characteristics ABMs can be useful to reproduce complex systems related to economics and social sciences [3]. However, given their peculiarity in dealing with the representation and simulation of complex systems, ABMs have been recently applied to solve optimization problems whose domains present several inter-related components in a distributed and heterogeneous environment ([4]; [5]), sometimes combined to other optimization techniques. In this context Davidsson et al. [6] state that Agent-Based simulation and conventional optimization techniques present complementary characteristics that should be integrated into hybrid approaches to solve complex problems, embedding optimization rules in the behavior of the agents.

Cardon et al. [7] propose an integration of genetic algorithm and multi-agent simulation for solving NP-hard scheduling problems. Wei et al. [8] develop an Agent-Based optimization framework for dynamic resource allocation problems; Desphande and Cagan [9] introduce an Agent-Based optimization algorithm for solving the process-planning problem that combines stochastic optimization techniques with knowledge-based search. Several applications of Agent-Based optimization to transportation problems can be found in the literature. Bocker et al. [10] utilize a multi-agent approach to cope with the train coupling and sharing problem on a railway transport system; Fernandez et al. [11] illustrate multi-agent service architectures for bus fleet management; Mes et al. [12] develop an Agent-Based approach for real-time transportation problems and compare it to more traditional methods. Barbucha et al. [13] and Barbucha et al. [14] introduce and test web-based architectures for solving optimization problems based on the utilization of agents' teams with common shared memory. An extensive review of Agent-Based approaches to solve optimization problems can be found in [15].

In this paper we illustrate an ABM model to solve a single location problem with "equity objectives".

The remainder of the paper is organized as follows. The following section provides a short presentation of location problems with "equity" objectives. Then, we illustrate an ABM to describe a single facility location problem with equity objectives and its implementation. Furthermore, a computational experience is provided, in order to test the validity of the proposed approach. Finally, some conclusions and directions for further researches are drawn.

2 Location Problems with Equity Objectives

A facility location problem consists of positioning facilities within a given space. The decision is made on the basis of an objective function, which can concern the minimization of costs or the maximization of benefits. In some contexts, the objective can be defined by a single criterion, while, in more complex situations, more criteria must be monitored simultaneously.

The so-called "balancing" objectives constitute a particular class of criteria in which the achievement of forms of "equity" in the access of potential users to the facility is pursued. A large number of equity measures have been provided in the extant literature for location problems. Most of them consider, as balancing objectives, measures of inequality in the distribution of the distances among the facilities and users (located in demand points).

According to Eiselt and Laporte [16], balancing objectives can be distinguished between those aiming at minimizing the spread of the deviation and those whose goal is the minimization of the deviation from target values. For instance, the minimization of the maximum distance or the minimization of the differences between the maximum and the minimum distance belong to the first set. On the other hand, minimizing the variance of the distances distribution or the maximum deviation from the average distance belongs to the second class of objectives. In addition, it is possible to distinguish between normalized and not normalized measures whether they are divided or not by a statistical measure of the distance distribution (for instance the mean). In the

normalized case the measure is dimensionless and its interpretation does not depend on the amplitude of the single instance. Surveys of the measures proposed in the literature are presented in [17] and [18] and, more extensively, in [16].

One of the most popular measures is represented by the Gini Coefficient. The Gini coefficient is related to the so called Lorenz curve which is often used by economists to describe the inequality in the wealth distribution across a population.

Traditionally, the abscissa measures the cumulative percentage of the population, the ordinate the cumulative percentage of the distribution of the wealth whose equity is to be calculated. The Lorenz curve has extremes at (0,0) and (1,1) with the most equitable distribution ($p\%$ of the population has $p\%$ of the good) being the straight 45° line that connects these two extremes. As inequity occurs, $p\%$ of the population holds less than $p\%$ of the good and the Lorenz curve drops below the straight 45° line. The Gini coefficient captures the magnitude of the inequity in a distribution as it is defined as the ratio of the area between the Lorenz curve and the straight “equity” line to the entire area below the equity line. In practice if the Gini coefficient is equal to 0, the distribution is the most equitable; on the other hand values of the coefficient close to 1 indicate distribution very inequitable.

Given the above definitions, a single-facility location problem can be solved in the plane by adopting the Gini Coefficient as objective function and looking for the position of a facility that minimizes the Gini coefficient of the distribution of the distances among the demand points and the facility itself. In particular, introducing a planar region and being

- n the number of demand points
- (x_i, y_i) the location of the demand point i for $i = 1, \dots, n$
- w_i the demand concentration for point i for $i = 1, \dots, n$
- (x_a, y_a) the unknown location of the facility on the whole plane.
- d_{ia} the distance between demand point i and the facility a for $i = 1, \dots, n$ where $d_{ia} = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$.

the problem will be aimed at minimize the function G_a (Gini coefficient). There are various ways of formulating the Gini coefficient; one of this describes it as exactly one half of the relative mean difference, which is defined as the arithmetic average of the absolute values of differences between all pairs of distances. For this reason it can be expressed as follows:

$$G_a = \frac{\sum_{i=1}^n \sum_{j=1}^n |d_{ia} - d_{ja}|}{2n \sum_{i=1}^n d_{ia}} \tag{1}$$

3 An Agent-Based Model for an Equity Location Problem

In the following we describe an ABM to solve a single facility location problem in which the objective is the minimization of the Gini coefficient. The model is an

adaptation of a general agent based framework proposed to tackle multi-facility location problems [19].

We suppose to locate a facility in a continuous space in which n demand points are positioned. In order to model through an agent based approach this environment, we consider a simulation space represented by a portion of plane (for instance a rectangle). In this space we consider the presence of two kinds of agents (see Figure 1):

- a set of “passive” agents representing the demand points;
- an “active” agent representing the facility to be located.

The two agent categories present different adaptive capabilities. Passive agents do not change position but they interact with the active agent in an autonomous way. They are neither re-active, as they do not react to any signal, nor pro-active as they do not pursue any objective. On the other hand, the active agent is both re-active, as it answers to the presence of passive agents, and pro-active, as it moves within the location space searching for positions according to a given objective.

We suppose that a force \underline{F}_{ia} operates on the active agent a , due to the presence of a demand point i . The intensity of the force is a function of the distance d_{ia} between i and a and it is considered significant only within a given distance R (i.e. $\underline{F}_{ia} > 0$ iff $d_{ia} \leq R$). By defining \underline{d}_{ai} as the position vector applied in a current position and pointing to i , a possible expression of the force, which pushes the agent a toward the position of i thanks to the demand w_i concentrated in i , can be defined as follows:

$$\underline{F}_{ia}^d = w_i \underline{d}_{ai} \quad \text{for } i : d_{ia} \leq R \tag{2}$$

This way, on the active agent a , there is a resulting force \underline{F}_{Ra} due to the simultaneous influence of the passive agents within the distance R (Figure 2). This force can be computed as follows:

$$\underline{F}_{Ra} = \sum_{i: d_{ia} \leq R} \underline{F}_{ia} \tag{3}$$

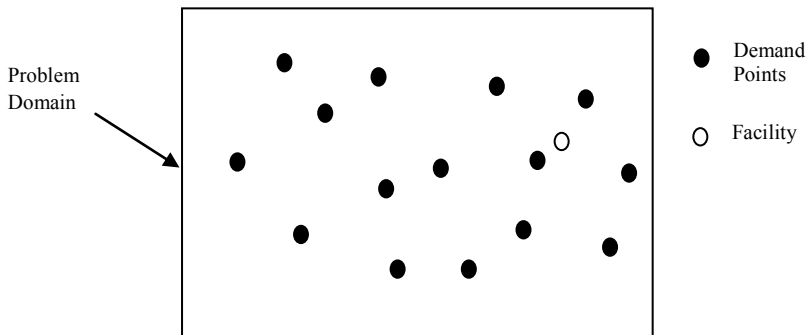


Fig. 1. Environment and agents for a single location facility problem

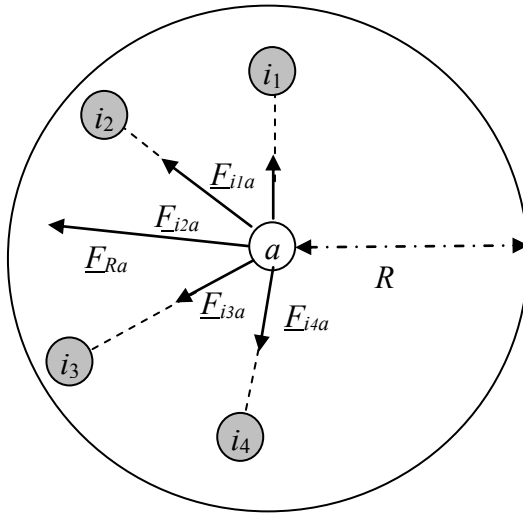


Fig. 2. The resulting force operating on the active agent

Basically, the force E_{Ra} will rule the search process within the location space. Driven by the force E_{Ra} , the facility agent will be pushed towards the areas within the location space in which the highest amount of demand is concentrated. Therefore, the agent will be exploring the areas of the location space identified by the direction of the resulting force vector E_{Ra} .

The illustrated model has been implemented within the *NetLogo* Agent-Based simulation environment (<http://ccl.northwestern.edu/netlogo>) using the proprietary programming language and its Java architecture. *NetLogo* allows reproducing the two agent categories introduced above. In particular, passive agents are represented by cells in a grid network, being each cell identified by a couple of integer coordinates.

The logical flow of the procedure is identified in Figure 3. The initial position of the active agent in the location space is randomly determined according to a uniform distribution with values ranging within the extreme coordinates of the location space. Then, at each iteration the resulting force is calculated by adding up the force that each single demand point is performing on the facility agent. Due to the action of this force, the active agent changes position and the solution assumes a new objective function value.

In practice the procedure behaves as a meta-heuristic, searching for better solutions thanks to a sort of evolutionary mechanism. After a given number of iterations that can be defined in terms of number of non-improving iterations, the procedure may restart from a new randomly generated initial solution (diversification process). The procedure stops after a given total number of evolution iterations or a fixed running time (stopping criterion).

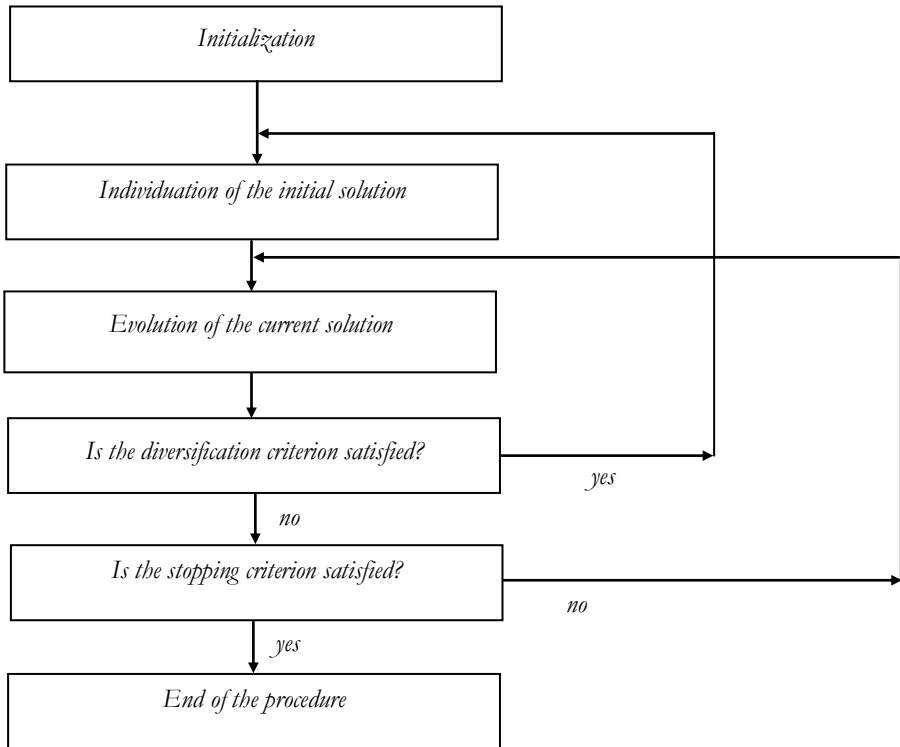


Fig. 3. Logical flow of the procedure

4 Computational Experiments

We tested the model on eight benchmark problems from 50 up to 10000 demand points in a unit square. These problems were randomly generated and used in [20] and [21]. In order to solve the instances, we used a 100x100 grid of agents, i.e., 10000 passive agents. As, in general, the position of an original discrete demand point did not coincide with any grid point, an adaptation of the instances demand data was performed, associating each demand point with the closest grid point; thus, each grid point has been weighted with a demand value equal to the number of associated demand points. The ABM provides the coordinates of the facility to be located with a ten-digit precision in the continuous location space. Then, the objective function value was computed as the variance of the distances of the original demand points from the located facility. This way, the objective function value includes the effects of the aggregation operation and, thus, associated errors [22].

Each problem was solved ten times using a maximum number of 150 iterations without any diversification process. Table 1 reports the minimum, the maximum and the average value of objective function; the latter is calculated as the mean of the

results obtained on ten experiments. Table 2 shows the gap of corresponding values obtained using the method illustrated in [21].

In particular we calculated the gap from minimum (GAP_{min}), the gap from maximum (GAP_{max}) and the gap from mean (GAP_{av}) as follows:

$$GAP_{min} = \frac{(ABM \text{ Best Sol.} - Drezner \text{ et al. (2009) Best Sol.})}{Drezner \text{ et al. (2009) Best Solution}} \cdot 100 \quad (4)$$

$$GAP_{max} = \frac{(ABM \text{ Worst Sol.} - Drezner \text{ et al. (2009) Worst Sol.})}{Drezner \text{ et al. (2009) Worst Sol.}} \cdot 100 \quad (5)$$

$$GAP_{av} = \frac{(ABM \text{ Average Value} - Drezner \text{ et al. (2009) Average Value})}{Drezner \text{ et al. (2009) Average Value}} \cdot 100 \quad (6)$$

The results show that our approach improves the best known solution just in one case (negative gap from the minimum). However the approach obtains significant improvement on the gap from maximum (6 cases on 8) and often (3 cases on 8) improvements on the average value. The results appear more interesting considering the computational times. Table 3 shows the comparison of the computational times for each test problem.

Our computational times are calculated on a PC with DualCoreT2250 2.0 Ghz CPU and 2 GB of RAM; those indicated for the procedure illustrated in [21] have been reported on this paper, considering the best choice of the calibration parameters and are calculated on a 2.8 GHz Pentium IV desktop computer with 256 MB RAM. We can observe that the computing times are worst for smaller problems but significantly better in the case of bigger problems.

This is substantially due to the fact that the computing times of the ABM based procedure depends on the number of demand points only for the calculation of the force. The times significantly depend on the number of iterations which are quite limited.

Table 1. Results for the Single Facility Minimum Gini Coefficient Problem

Demand Points	Minimum	Maximum	Mean
50	0.181624631	0.199905466	0.194603937
100	0.183210487	0.19021816	0.18819192
200	0.194902419	0.195462255	0.195226727
500	0.203111338	0.203656773	0.203386269
1000	0.212047566	0.212105634	0.212080746
2000	0.214517294	0.214809269	0.214634729
5000	0.213462441	0.213488354	0.213472828
10000	0.213968594	0.214003338	0.213983313

Table 2. Comparison results for the Single Facility Minimum Gini Coefficient Problem

Demand Points	GAP_{min}	GAP_{max}	GAP_{av}
50	11.70 %	-2.25%	4.18%
100	-0.16 %	-13.06%	-6.23%
200	2.20 %	-9.04%	-5.55%
500	6.12 %	-4.43%	-0.64%
1000	5.76 %	-0.09%	2.40%
2000	5.21 %	-0.46%	1.48%
5000	1.94 %	0.13%	1.03%
10000	2.52 %	0.85%	1.32%

Table 3. Computational times for the Single Facility Minimum Gini Coefficient Problem

Computational Times (sec.)		
Demand Points	Our Approach	Drezner et al. (2009)
50	70	0.15
100	70	0.34
200	70	1.14
500	75	4.10
1000	100	15.92
2000	115	71.47
5000	115	585.02

5 Conclusions

In this paper, we have presented an ABM for modeling and solving a location problem with the minimization of one of the most used equity objectives the Gini coefficient. Besides its description, we have presented some computational results in which the results provided by the model are compared with others proposed in literature. The results have shown that the model is able to find good quality solutions within very limited computational times.

These results encourage further developing the research on the application of ABMs to solve combinatorial optimization problems and location problems as well. Future researches will be devoted to extend the class of treatable equity location problems and to determine the effect of aggregation errors [22] on the quality of the solution.

References

1. Wooldridge, M., Jennings, N.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995)
2. Billari, F.G., Fent, T., Prskawetz, A., Scheffran, J. (eds.): Agent-Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences (Contributions to Economics). Physica, Heidelberg (2006)

3. Conte, R., Hegselmann, R., Terna, P. (eds.): *Simulating Social Phenomena*. Springer, Berlin (1997)
4. Weiss, G.: *Multiagent Systems, a modern approach to distributed artificial intelligence*. MIT Press, Cambridge (1999)
5. Wooldridge, M.: *An Introduction to Multiagent Systems*. John Wiley and Sons, New York (2002)
6. Davidsson, P., Johansson, S., Persson, J., Wernstedt, F.: *Combining Agent-Based approaches and Optimization Techniques*. In: *Proceedings of the EUMAS 2003 Conference* (2003)
7. Cardon, A., Galinho, T., Vacher, J.P.: *Genetic algorithms using multi-objectives in a multi-agent system*. *Robotics and Autonomous Systems* 33(2), 179–190 (2000)
8. Wei, Q., Sawaragi, T., Tian, Y.: *Bounded optimization of resources allocation among multiple agents using an organizational decision model*. *Advanced Engineering Informatics* 19(1), 67–78 (2005)
9. Desphande, S., Cagan, J.: *An Agent Based Optimization Approach to Manufacturing Process Planning*. *ASME Journal of Mechanical Design* 126(1), 46–55 (2004)
10. Bocker, J., Lind, J., Zirkler, B.: *Using a multi-agent approach to optimise the train coupling and sharing system*. *European Journal of Operational Research* 134(1), 242–252 (2001)
11. Fernandez, A., Alonso, E., Ossowski, S.: *A multi-agent service architectures for bus fleet management*. *Integrated Computer-Aided Engineering* 11(2), 101–115 (2004)
12. Mes, M., van der Heijden, M., van Harten, A.: *Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems*. *European Journal of Operational Research* 181(1), 59–75 (2007)
13. Barbucha, D., Czarnowski, I., Jedrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: *e-JABAT - An Implementation of the Web-Based A-Team*. In: Nguyen, N.T., Jain, I.C. (eds.) *Intelligent Agents in the Evolution of Web and Applications*. SCI, pp. 57–86. Springer, Heidelberg (2009)
14. Barbucha, D., Czarnowski, I., Jedrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: *Influence of the Working Strategy on A-Team Performance*. In: Szczerbicki, E., Nguyen, N.T. (eds.) *Smart Information and Knowledge Management*. SCI, vol. 260, pp. 83–102. Springer, Heidelberg (2010)
15. Barbati, M., Bruno, G., Genovese, A.: *Applications of Agent-Based Models for Optimization Problems: a literature review*. In: Collan, M. (ed.) *Proceedings of the 2nd International Conference on Applied Operational Research*. *Lecture Notes in Management Science*, vol. 2, Tadbir Institute for Operational Research, Systems Design and Financial Services, Teheran (2010)
16. Eiselt, H.A., Laporte, G.: *Objectives in Location Problems*. In: Drezner, Z. (ed.) *Facility Location: A Survey of Applications and Methods*, pp. 151–180. Springer, Berlin (1995)
17. Erkut, E.: *Inequality measures for location problems*. *Location Science* 1, 199–217 (1993)
18. Marsh, M.T., Schilling, D.A.: *Equity measurement in facility location analysis: A review and framework*. *European Journal of Operations Research* 74, 1–17 (1994)
19. Bruno, G., Genovese, A., Sgalambro, A.: *An Agent-Based framework for modeling and solving location problems*. *TOP* 18, 81–96 (2010)
20. Drezner, T., Drezner, Z.: *Equity Models in Planar Location*. *Computational Management Science* 4, 1–16 (2007)
21. Drezner, T., Drezner, Z., Guyse, J.: *Equitable service by a facility: Minimizing the Gini coefficient*. *Computers and Operations Research* 36, 3240–3246 (2009)
22. Plastria, F.: *New error bounds in continuous minisum location for aggregation at the gravity centre*. *Studies in Locational Analysis* 14, 101–119 (2000)

Rotation Forest with GEP-Induced Expression Trees

Joanna Jędrzejowicz¹ and Piotr Jędrzejowicz²

¹ Institute of Informatics, Gdańsk University,
Wita Stwosza 57, 80-952 Gdańsk, Poland
jj@inf.ug.edu.pl

² Department of Information Systems, Gdynia Maritime University,
Morska 83, 81-225 Gdynia, Poland
pj@am.gdynia.pl

Abstract. In this paper we propose integrating two techniques used in the field of the supervised machine learning. They include rotation forest and gene expression programming. The idea is to build a rotation forest based classifier ensembles using independently induced expression trees. To induce expression trees we apply gene expression programming. The paper includes an overview of the proposed approach. To evaluate the approach computational experiment has been carried out. Its results confirm high quality of the proposed ensemble classifiers integrating rotation forest with gene expression programming.

Keywords: gene expression programming, rotation forest algorithm, ensemble classifiers.

1 Introduction

Classifier ensembles are constructed through somehow merging classifiers in order to achieve superior accuracy compared to that of single classifiers. A single classifier can be viewed as a simple agent capable of predicting a class label of an object upon being presented with its attribute values. Among well known classifier ensemble models one can mention AdaBoost, Bagging, Random Forest and Rotation Forest. AdaBoost [5] is an algorithm for constructing an ensemble classifier as a linear combination of simple weak classifiers. Bagging [2] is a method for generating multiple versions of a classifier and using these to get an ensemble classifier. The aggregation averages over the versions when predicting a numerical outcome and does a majority vote when predicting a class. Random Forest [3] is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Rotation Forest method introduced by [10] is an extension of the Random Forest idea. The base classifiers are also independent but each weak classifier is trained on the whole data set in a rotated feature space.

Gene expression programming introduced by Ferreira [4] is an automatic programming approach. In GEP computer programs are represented as linear character strings of fixed-length called chromosomes which, in the subsequent fitness

evaluation, can be expressed as expression trees of different sizes and shapes. The method has flexibility and power to explore the entire search space, which comes from the separation of genotype and phenotype.

Several approaches to constructing ensemble classifiers from expression trees induced using gene expression programming were proposed in earlier papers of the authors and presented in [6], [7], [8]. In this paper the idea of using expression trees induced through applying gene expression programming as the base classifiers for the Rotation Forest method, is explored. The paper is organized as follows. In section 2 classifiers induced by gene expression programming are proposed and discussed. In section 3 the Rotation Forest method is overviewed and the ensemble classifier integrating expression trees within Rotation Forest is described. In section 4 the results of the validating experiment are shown. Finally, section 5 contains conclusions.

2 Using Gene Expression Programming to Induce Classifiers

In this paper the general data classification problem is considered. In what follows C is the set of categorical classes which are denoted $1, \dots, |C|$. We assume that the learning algorithm is provided with the learning instances $LD = \{ \langle d, cl \rangle \mid d \in D, cl \in C \} \subset D \times C$, where D is the space of attribute vectors $d = (w_1^d, \dots, w_N^d)$ with w_i^d being symbolic or numeric values, N - the number of attributes. The learning algorithm is used to find the best possible approximation \bar{f} of the unknown function f such that $f(d) = cl$. Then \bar{f} can be used to find the class $\bar{cl} = \bar{f}(\bar{d})$ for any $\bar{d} \in D - (LD|D)$. The set of learning instances LD consists of two subsets $LD = TD \cup TS$ that is TD -training set and TS - testing set.

As usual when applying GEP methodology, the algorithm uses a population of chromosomes, selects them according to fitness and introduces genetic variation using several genetic operators. Each chromosome is composed of a single gene (chromosome and gene mean the same in what follows) divided into two parts as in the original head-tail method [4]. The size of the head (h) is determined by the user with the suggested size not less than the number of attributes in the dataset. The size of the tail (t) is computed as $t = h(n - 1) + 1$ where n is the largest arity found in the function set. In the computational experiments the functions are: logical AND, OR, XOR, NOR and NOT. Thus $n = 2$ and the size of the gene is $h + t = 2h + 1$. The terminal set contains triples $(op, attrib, const)$ where op is one of relational operators $<, \leq, >, \geq, =, \neq$, $attrib$ is the attribute number, and finally $const$ is a value belonging to the domain of the attribute $attrib$. As usual in GEP, the tail part of a gene always contains terminals and head can have both, terminals and functions. Observe that in this model each gene is syntactically correct and corresponds to a valid expression. Each attribute can appear once, many times or not at all. To introduce variation in the population the following genetic operators are used:

- mutation,
- transposition of insertion sequence elements (IS transposition),

- root transposition (RIS transposition),
- one-point recombination,
- two-point recombination.

During GEP learning, the individuals are selected and copied into the next generation based on their fitness and the roulette wheel sampling with elitism which guarantees the survival and cloning of the best gene in the next generation. The details of the applied GEP learning can be found in [6], [7].

The learning stage is class-specific:

- the fitness function is class dependent: for class cl and gene g the value of $fitness^{cl}(g)$ is defined as the difference between the number of rows from class cl for which g is 'true', and the number of rows from classes $\neq cl$ for which g is 'true',
- learning takes place separately for each class $cl \in C$.

The algorithm of GEP learning is shown as Algorithm 1.

Algorithm 1. Algorithm GEP learning

Require: class cl , training data, integer NG - number of iterations

Ensure: gene g best fitting class cl

- 1: start with a random population Pop of genes
 - 2: **for** $i = 1$ to NG **do**
 - 3: express genes as expression trees,
 - 4: calculate fitness of each gene,
 - 5: keep best gene
 - 6: **for all** $g \in Pop$ **do**
 - 7: select genes
 - 8: mutation
 - 9: transposition of insertion sequence elements (IS transposition)
 - 10: root transposition (RIS transposition)
 - 11: one-point recombination
 - 12: two-point recombination
 - 13: **end for**
 - 14: **end for**
 - 15: calculate fitness and keep the best gene g
-

3 Principal Component Analysis and Rotation Forest Ensemble Method

Rotation Forest method creates an ensemble of base classifiers, each based on a different axis rotation of attributes. To create the training dataset, the set of attributes is randomly split into a given number of subsets and Principal Component Analysis (PCA) is applied to each subset. The algorithm for PCA is given in Algorithm 2. Let M stand for the number of data points, each with N attributes.

Algorithm 2. Algorithm PCA

Require: dataset - a matrix X consisting of M rows and N columns

Ensure: the set of M eigenvectors

- 1: transform the data matrix X into a dataset whose mean is 0 - calculate the mean value for each column and subtract the mean from each each row in the given column
 - 2: find the covariance matrix $Cov = \frac{1}{N-1} X \cdot X^T$
 - 3: compute M eigenvectors of the covariance matrix Cov
-

Algorithm 3. Algorithm Rotation Forest with GEP – RF-GEP learning

Require: TD training dataset with M points, each described by N attributes (TD is a matrix of size $M \times N$), Y - correct class labels of the training dataset TD (Y is a matrix of size $N \times 1$ of elements from C), L - the number of classifiers in the ensemble, K - the number of subsets

Ensure: population of L gene expressions for each class and L rotation matrices for each class

- 1: **for all** $cl \in C$ **do**
 - 2: **for** $i = 1$ to L **do**
 - 3: split the attribute set F into K subsets $\{F_{ij}\}_{j \leq K}$ each with M_j^i attributes
 - 4: **for** $j = 1$ to K **do**
 - 5: let X_{ij} be the dataset TD for the attributes in F_{ij}
 - 6: transform X_{ij} by deleting 25 % rows, randomly
 - 7: apply algorithm PCA to the transformed X_{ij} to obtain eigenvectors $\mathbf{a}_{ji}^1, \dots, \mathbf{a}_{ji}^{M_j^i}$
 - 8: **end for**
 - 9: arrange the eigenvectors in matrix E_i^{cl} as shown in (II)
 - 10: construct the rotation matrix R_i^{cl} from E_i^{cl} by rearranging the columns to match the order in F
 - 11: **end for**
 - 12: use algorithm GEP to find the best gene for the training dataset $(TD \cdot R_i^{cl}, Y)$
 - 13: **end for**
-

The matrix with eigenvectors used in Algorithm 3, is defined in (II) (note that the elements should contain the subscript cl - it is omitted for clarity).

$$E_i^{cl} = \begin{bmatrix} \mathbf{a}_{1i}^1 & \mathbf{a}_{1i}^2 & \dots & \mathbf{a}_{1i}^{M_1^1} & [0] & \dots & [0] \\ & [0] & & \mathbf{a}_{2i}^1 & \mathbf{a}_{2i}^2 & \dots & \mathbf{a}_{2i}^{M_2^1} & \dots & [0] \\ & \vdots & & \vdots & & \ddots & & \vdots & \\ [0] & & [0] & & & \dots & \mathbf{a}_{Ki}^1 & \mathbf{a}_{Ki}^2 & \dots & \mathbf{a}_{Ki}^{M_K^1} \end{bmatrix} \quad (1)$$

In the testing stage, for each point (x, cl) from the testing dataset TS , the majority vote is used for the data vector x modified by the rotation matrix to find the class \overline{cl} . The final result of the classification is the number of correctly

Algorithm 4. Algorithm Rotation Forest with GEP – RF-GEP testing

Require: $TS = \{(x_j, cl_j) : j = 1, \dots, Mt, cl_j \in C\}$ testing dataset with Mt points, each described by N attributes, population of L gene expressions g_i^{cl} for each class and L rotation matrices R_i^{cl} for each class cl

Ensure: qc - quality of classification

```

1:  $qc = 0$ 
2: for all  $(x, cl) \in TS$  do
3:   let  $\bar{cl} = \text{argmax}_{cl \in C} (\sum_{i=1}^L sg(g_i^{cl}(x \cdot R_i^{cl})))$ 
4:   if  $cl = \bar{cl}$  then
5:      $qc = qc + 1$ 
6:   end if
7: end for
8:  $qc = qc/Mt$ 

```

classified vectors divided by the size of the testing dataset. The testing stage for RF-GEP is defined in Algorithm 4. We use the notation

$$sg(\varphi) = \begin{cases} 1 & \text{if } \varphi \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

The proposed approach preserves main features of the original Rotation Forest method. As it was stressed in [9] the main heuristic is to apply feature extraction and to subsequently reconstruct a full feature set for each classifier in the ensemble. This is done through dividing randomly the feature set into a number of subsets, applying principal component analysis to each subset and constructing new set of features by combining all principal components. However the base classifiers are not decision trees but expression trees specifically induced using the gene expression programming technique.

4 Computational Experiment Results

To evaluate the proposed approach computational experiment has been carried out. The experiment involved the following datasets from the UCI Machine Learning Repository [1]: Wisconsin Breast Cancer (WBC), Diabetes, Sonar, Australian Credit (ACredit), German Credit (GCredit), Cleveland Heart (Heart), Hepatitis and Ionosphere. Basic characteristics of these sets are shown in Table 1. In the reported experiment the following classification tools have been used: Rotation Forest with gene expression programming induced expression trees (RF-GEP) proposed in this paper, Cellular GEP with AdaBoost (GEP-ADA) proposed in [8], versus 16 well-known classifiers from WEKA Environment for Knowledge Analysis v. 3.7.0 [11] including classic Rotation Forest, Nave Bayes, Bayes Net, Logistic Regression, Radial Basis Function Network, Adaboost, Support Vectors Machine, Ensemble Selection, Bagging, Classification via Clustering, Random Committee, Decision Table, FT Tree, Random Forest and C4.5.

Table 1. Datasets used in the computational experiment

name	data type	attribute type	no. instances	no. attributes
WBC	multivariate	integer	699	11
Diabetes	multivariate, time-series	categorical, integer	768	9
Sonar	multivariate	real	208	61
ACredit	multivariate	categorical, integer, real	690	15
GCredit	multivariate	categorical, integer	1000	21
Heart	multivariate	categorical, real	303	14
Hepatitis	multivariate	categorical, integer, real	155	20
Ionosphere	multivariate	integer, real	351	35

Table 2. Comparison of the classifier accuracy (%).

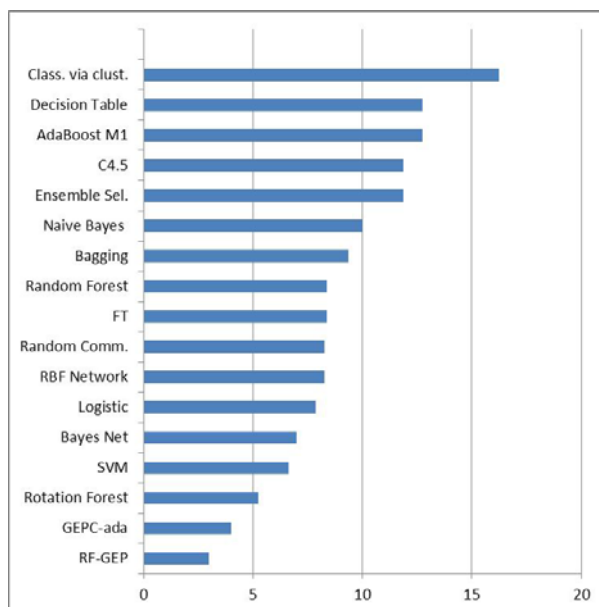
no.	classifier	WBC	Diab.	Sonar	ACr.	GCr.	Heart	Hep.	Ion.	Mean
1	RF-GEP	97,65	76,99	80,79	88,04	76,27	82,35	86,46	93,73	85,29
2	cGEP-ada	95,86	77,21	81,24	86,52	77,37	83,84	87,13	91,35	85,07
3	Rotation Forest	96,99	76,69	84,13	87,25	74,80	80,74	82,58	93,73	84,61
4	Random Committee	95,99	73,95	84,13	83,48	73,90	80,37	84,52	92,59	83,62
5	SVM	96,99	77,34	75,96	84,92	75,10	84,07	85,16	88,60	83,53
6	Bayes Net	97,14	74,35	80,28	86,23	75,50	81,11	83,22	89,46	83,41
7	FT	96,99	77,34	79,81	85,51	68,30	82,96	81,29	90,31	82,81
8	Logistic	96,56	77,21	73,08	85,22	75,20	83,70	82,58	88,89	82,81
9	Random Forest	96,13	73,82	80,77	85,07	72,50	78,15	82,58	92,87	82,74
10	Bagging	95,56	74,61	77,40	85,07	74,40	79,26	84,52	90,88	82,71
11	RBF Network	95,85	75,39	72,11	79,71	74,00	84,07	85,80	92,31	82,41
12	Ensemble Selection	94,42	74,61	75,48	84,93	73,10	80,00	81,29	90,59	81,80
13	AdaBoost M1	94,85	74,34	71,63	84,64	69,50	80,00	82,58	90,88	81,05
14	C4.5	94,56	73,82	71,15	86,09	70,50	76,66	83,87	91,45	81,01
15	Naive Bayes	95,99	76,30	67,78	77,68	75,40	83,70	84,51	82,62	80,50
16	Decision Table	95,28	71,22	69,23	83,48	71,00	84,81	76,13	89,46	80,08
17	Class. via clustering	95,71	64,84	54,32	74,06	56,60	77,04	74,19	70,94	70,96

Computations involving RF-GEP have been run with the following settings: number of features in a subset 3, number of classifiers in the ensemble 10, GEP population size 50, number of iterations in GEP 100, probabilities of mutation, RIS transposition, IS transposition, 1-point recombination and 2-point recombination - all set to 0.2. Parameters in GEPC-ADA has been set as reported in [8]. In all WEKA classifiers the default parameter settings have been used.

Tables 2, 3 show computation results averaged over 10 repetitions of the 10-cross-validation scheme. Performance measures include classifier accuracy shown in Table 2 and the area under the ROC curve calculated as the Wilcoxon-Mann-Whitney statistic shown in Table 3.

Table 3. Comparison of the area under the ROC curve

no.	classifier	WBC	Diab.	Sonar	ACr.	GCr	Heart	Hep.	Ion.	Mean
1	Bayes Net	0,992	0,806	0,877	0,921	0,780	0,902	0,882	0,948	0,889
2	Rotation Forest	0,986	0,810	0,925	0,921	0,757	0,883	0,826	0,967	0,887
3	GEPC-ada	0,978	0,826	0,845	0,893	0,877	0,843	0,798	0,969	0,884
4	Random Committee	0,987	0,785	0,912	0,900	0,761	0,866	0,848	0,976	0,880
5	RF-GEP	0,983	0,812	0,896	0,905	0,803	0,842	0,821	0,962	0,878
6	Random Forest	0,988	0,779	0,912	0,914	0,748	0,854	0,835	0,956	0,877
7	Bagging	0,987	0,817	0,873	0,915	0,776	0,875	0,821	0,934	0,877
8	Naive Bayes	0,986	0,819	0,800	0,876	0,787	0,898	0,860	0,925	0,873
9	AdaBoost M1	0,989	0,801	0,841	0,929	0,723	0,878	0,851	0,944	0,871
10	Logistic	0,993	0,832	0,763	0,904	0,785	0,900	0,802	0,870	0,857
11	RBF Network	0,986	0,783	0,809	0,854	0,753	0,893	0,835	0,939	0,856
12	Ensemble Selection	0,987	0,806	0,824	0,906	0,745	0,852	0,733	0,940	0,854
13	Decision Table	0,987	0,773	0,763	0,909	0,723	0,877	0,763	0,895	0,830
14	FT	0,975	0,763	0,846	0,883	0,662	0,857	0,754	0,896	0,830
15	SVM	0,968	0,720	0,758	0,856	0,671	0,837	0,756	0,853	0,805
16	C4.5	0,955	0,751	0,743	0,887	0,639	0,744	0,708	0,892	0,798
17	Class. via clustering	0,949	0,605	0,543	0,749	0,524	0,772	0,803	0,715	0,710

**Fig. 1.** The average Friedman ranks for the classification accuracy of different classifiers

To evaluate the performance of the RF-GEP approach, the Friedman's non-parametric test using ranks of the data has been applied. For each of the two - classification accuracy and ROC area the following hypotheses were tested:

- Null Hypothesis H0: All of the 17 population distribution functions are identical.
- Alternative Hypothesis H1: At least one of the populations tends to yield larger observations than at least one of the other populations.

Analyses of the experiment results shows that in each case, that is for the population of the classification accuracy observations and the population of the ROC area observations, the null hypothesis should be rejected at the significance level of 0,05. The average Friedmans ranks for the classification accuracy are shown in Fig. 1.

5 Conclusions

The paper proposes an approach based on integrating the Rotation Forest method with Gene Expression Programming to construct ensemble classifiers. Validating experiment results allow to draw the following conclusions:

- Using GEP-induced expression trees as base classifiers within the Rotation Forest method results in constructing high quality ensemble classifiers from the point of view of the classification accuracy.
- The proposed approach could be competitive in comparison with a majority of standard classifiers.
- High classification accuracy of RF-GEP ensembles does not necessary assures best performance from the point of view of the area under the ROC curve.

Future research should focus on increasing efficiency of the approach through possibly diversification of tools for inducing simple agents serving as base classifiers.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. School of Information and Computer Science. University of California, Berkeley(2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Breiman, L.: Bagging Predictors. *Machine Learning* 24, 123–140 (1996)
3. Breiman, L.: Random Forests. *Machine Learning* 45, 5–32 (2001)
4. Ferreira, C.: Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. *Complex Systems* 13(2), 87–129 (2001)
5. Freund, Y., Schapire, R.: A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
6. Jędrzejowicz, J., Jędrzejowicz, P.: GEP-induced expression trees as weak classifiers. In: Perner, P. (ed.) *ICDM 2008*. LNCS (LNAI), vol. 5077, pp. 129–141. Springer, Heidelberg (2008)

7. Jędrzejowicz, J., Jędrzejowicz, P.: A Family of GEP-Induced Ensemble Classifiers. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 641–652. Springer, Heidelberg (2009)
8. Jędrzejowicz, J., Jędrzejowicz, P.: Cellular GEP-Induced Classifiers. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS, vol. 6421, pp. 343–352. Springer, Heidelberg (2010)
9. Kuncheva, L.I., Rodríguez, J.J.: An Experimental Study on Rotation Forest Ensembles. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 459–468. Springer, Heidelberg (2007)
10. Rodriguez, J.J., Kuncheva, I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1630 (2006)
11. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Combined Course Programmes Generation in Multi-agent e-Learning System Using Policy-Based HTN Planning

Pavel Nikolaev and Aladdin Ayesh

De Montfort University, Leicester, UK
{pn, aayesh}@dmu.ac.uk

Abstract. As integration processes have been intensified in the area of higher education, they should be supported by information technologies. A Combined Course Programmes (CCP) generation mechanism based on HTN planning was developed for a multi-agent system, managing the distributed e-Learning environment. The limitations of HTN planning were overcome by XACML policy language, that was used for specification of preconditions. This improves the expressiveness and facilitates planning in a multi-domain environment supported by different experts. In order to improve the effectiveness of HTN planning in the multi-domain hierarchical environment, a domain refinement technique was designed. Its efficiency gains were confirmed by experiments.

Keywords: Multi-agent e-Learning system, Hierarchical Task Network (HTN) planning, policy specification language, refinement search.

1 Introduction

During the several last years there has been a trend for integration in the areas of distant learning and academic education. Different Learning Management Systems and Learning Objects Repositories are being united into integrated e-Learning environments. In the area of higher education student mobility is motivated within the Bologna process. Student mobility is a possibility of a student to change universities several times during his educational course. During this process credits from different universities should be accepted. CCPs are developed by several universities in common, they describe an educational track when student mobility is applied. Moreover, on graduation students can get joint or double diploma. CCPs are developed based on the structure and content requirements using components of existing Course Programmes (semesters and modules). We'll refer to these components as Learning Objects (LO).

Agents and multi-agent systems (MAS) are effectively used for a distributed e-Learning environment management. Their core function is to integrate educational resources and services of different organisations for their effective usage from any point of the environment [1]. MAS technology provides the means for an effective search and retrieval of the educational content from different repositories, for a flexible management of heterogeneous rapidly changing e-Learning

environment. It was proposed to introduce the CCP development function into these MAS to support the educational integration processes [7].

In this paper it's proposed to use Hierarchical Task Network (HTN) planning for CCP generation in the multi-agent e-Learning environment. A motivation of this work is given in section 2. HTN concepts are described in section 3. The limitations of HTN planning for CCP generation were overcome using a novel approach, when XACML policy language is used for specification of preconditions (Section 4). The overall architecture of the multi-agent e-Learning system is presented in section 5.1. A planning agent of this system realises the CCP generation functionality described in section 5.2. One of the challenges of the CCP generation is a computational complexity of planning in large environments with rigid constraints. The descending policy evaluation mechanism was proposed to improve its effectiveness (Section 6). This approach was realised using a set of fine-grained methods. This provides a possibility to reason about LO variable instantiation on a higher (domain) level. These methods form a new domain refinement operation, that should be considered as an alternative to task decomposition operation during the planning. Finally, a set of evaluation tests were carried out, indicating the efficiency of HTN planning with domain refinement technique for complex tasks (Section 7).

2 Motivation

E-Learning systems in current use have two types of functions that can be utilised for the CCP generation. These are effective, context-aware, personalised search for LOs throughout the distributed environment and Curriculum Generation (CG), that is a part of Intelligent Tutoring Systems (ITS). CG is a generation of detailed electronic LOs using rigidly described pieces of the educational content within some knowledge area based on the student requirements, peculiarities and abilities. These functions along can't cover the CCP generation function.

When student mobility in a higher education is considered, the aim of the student isn't only to learn some topic or material. He needs to study in a set of universities in order to get a concrete degree(s). This implies that a great attention should be paid to a compliance with a broad range of regulations in the educational environment during the CCP generation. These regulations determine student transfer rules, admission and credits acceptance rules, requirements to CCPs, degrees and a structure of the educational tracks, etc. They can be specified on different levels of the educational system: university, region, country, etc. They are different in different domains, and they can be changed on a regular basis. Moreover, requirements to a source of the educational content and to a structure of CCP are parts of the requirements to a new CCP also.

3 Hierarchical Task Network (HTN) Planning

HTN planning is built on top of the classical planning [4]. The core element of HTN planning is a *task network* $TN = \langle T, O \rangle$, where T is a set of tasks and O is

a set of ordering constraints on these tasks. Set T consists of two types of tasks: primitive tasks, corresponding to planning operators (actions), and compound tasks. Compound tasks are decomposed into new task networks by application of methods. *Method* is a tuple $\langle T, C, TN \rangle$, that encodes a domain-specific rule that task T can be decomposed into task network TN , when constraints C are satisfied (including preconditions constraints). A decomposition of high-level tasks using specially designed domain-specific methods is the main mechanism assuring a high efficiency of HTN planning. *Problem*, that should be solved by HTN planner, is specified as a tuple of an initial state and a task network, that should be carried out.

4 Policy-Based HTN Planning Approach

Based on analysis of the educational regulations, following limitations of HTN planning were revealed. These regulations include numeric computations, aggregative operations and quantifiers. It's impossible to specify them using ordinal preconditions of HTN, since they are based on logic (the same limitation of HTN planning is stated in [6]). In the multi-domain environment, considered in this paper, a precondition of each action and method should reflect all corresponding regulations from several domains of the environment. This leads to really comprehensive preconditions that are difficult to specify, test and support, since they are specified as massive logical expressions. This complicates a task of HTN environment specification, that is rather complex by itself [4]. Moreover, preconditions are maintained by different experts in this environment, so they can be changed independently. This leads to a service disruption, as HTN planning environment should be recompiled after each modification.

A new policy-based planning approach was introduced to overcome these limitations. Policies, specified using a policy language, are used as extra means for specification of preconditions. Policy based management is a well-recognised way to facilitate management of heterogeneous systems under complex dynamic rules [8]. Policy is a set of simple rules, that determine behaviour of the system in a concrete situation (or from some point of view). But policies are compositional, several simple policies can be composed to form a more complex policy. This simplifies policy support and specification, provides several persons with a possibility to support policies independently. Moreover, policies are descriptive and can be changed while the system executes without the need to recompile it.

XACML policy language, that was used for the specification of preconditions, is a standard access control (AC) policy language [3]. XACML has the following extra advantages: extremely high expressiveness (numeric computations, string transformations, aggregative functions, quantifiers and XPath expressions are supported), graphical tools for policy authoring, multi-valued attributes.

AC policy is a set of rules that determine, whether a subject *Sub* can be authorised to execute some action *Act* on a resource *Res* in a specific situation *Env*. Policy requests contain all the attributes of *Sub*, *Act*, *Res* and *Env*, that are used to infer a decision (“deny”, “accept” or “indeterminate”) using policy rules.

For CCP generation LOs are used as resources, actions correspond to planning operators and subject is a student. All actions, that should be executed in order to carry out a new CCP, should be evaluated against the policy. Special function *PolicyEval* (*SubID*, *ResID*, *ActID*, *Par1*, ..., *Par2*) is added to every action in order to generate and evaluate policy requests. Action parameters *Par_i* and identifiers of *Sub* and *Res* are determined using ordinal preconditions. Subject and recourse attributes are extracted from the planner's world state. Examples of policy rules for CCP generation are following: "A CCP can't be more than 1 year longer than the CP with the same diploma", "A student studying according to the CP_1 can change his programme to the CP_2 , in case he had already studied 80% of the CP_2 during the CP_1 ".

5 CCP Generation in Multi-agent e-Learning System

5.1 Multi-agent e-Learning System for CCP Generation

In a distributed e-Learning environment, managed by agents, each university is represented by a service provider agent. University's LOs represent its possibilities. They are stored in a university's repository. A "Student" agent receives requirements to LOs from the user. It's aim is to find LOs, matching the requirements most, through the communication with other agents.

When a CCP is requested from the system, university agents should share their possibilities to achieve the aim of the system: to generate a CCP satisfying the requirements provided by the user. This task is carried by a planning agent, that uses HTN planning to generate the CCP, combining LOs of different agents based on requirements to the CCP. The scheme of multi-agent environment for e-Learning, where CCPs should be generated, is shown at figure 1. The planning agent should guarantee that the CCP conforms to all the policies applicable to it. Policies are divided into two sets: university policies and global policies (for higher-level domains). University policies are managed by university agents. Global policies are stored in several global repositories, corresponding to different high-level domains in the environment. Global policy mediator is responsible for policy retrieval based on planning agent's requests.

This system can be used in distant learning, as well as in traditional education. LOs can be accessed from different repositories by the student according to the CCP generated, or CCP can be used to manage an educational process. In this paper we examine the CCP generation function of the planning agent. We consider that it has all the information and policies required for CCP generation.

5.2 CCP Generation as a HTN Planning Task

Planning environment for CCP generation is a multi-domain hierarchical environment, formed by a domain tree $\mathcal{G} = \langle L, E \rangle$. $L = \{L_x^y\}$ is a set of domains (e.g., "country", "university", etc.), organised hierarchically according to E (x - level of the domain in a tree, y - number of the domain on this level). Each domain is a set of LOs. Each LO should be included in one and only leaf-domain. Any

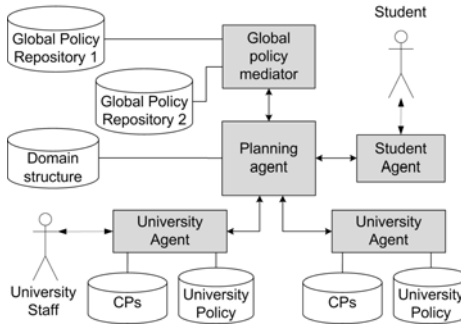


Fig. 1. Multi-agent environment for CCP generation

domain contains all LOs of its descendant domains. They are managed by a policy of this domain. So policies for hierarchically organised domains are combined using conjunction (a policy request must be accepted in all ancestor domains).

Plan that is generated by a HTN algorithm is a sequence of actions that are carried out over different LOs. These LOs form a new CCP. The correctness of action sequence is guaranteed by HTN operators and methods, encoding experts' knowledge about possible scheme of education. From the solution plan a track can be extracted. Track is a sequence of CP or $|CP|$ ($|CP|$ is an interval of CP , e.g. a sub-sequence of terms in CP). Each CP or $|CP|$ corresponds to one slot of the track. As a special part of the planning problem a Partial track (PT) for the solution is defined. PT specifies user's requirements to the structure of CCP. PT is a sequence of domains $PT = \langle L_{x1}^{y1}, \dots, L_{xn}^{yn} \rangle$. Each domain corresponds to one slot of the track and constrains a set of LOs that can be used in it ($|CP|_i \subseteq L_{xi}^{yi}$).

6 Domain Refinement for Policy-Based HTN Planning

6.1 Descending Policy Evaluation

One of the crucial difficulties of planning in the multi-domain hierarchical environment is a consistent choice of LOs, because of a large amount of them and because policies impose restrictions on their consistent use. The average number of possible LOs for LO-variable instantiation is much bigger than the average number of methods applicable to some task and average number of ways for constraint enforcement. According to a FAF (Fewest Alternatives First) strategy [9], late LO-variable instantiation approach should be applied. Tasks expansion and other variables instantiation should have a priority during the planning process.

Considering a hierarchical organisation of policies a *descending policy evaluation* approach can improve the effectiveness of planning. It uses a partial policy request, that is generated without a LO identifier using just properties of LO, extracted from the constraints for this LO $C_{LO} \subset C$. Such request can be evaluated into "Indeterminate" result when the required information is missing.

At step 1 of descending policy evaluation a partial policy request is evaluated in the domain, which is a PT restriction of the corresponding slot(s). During

step 2 the partial request is iteratively evaluated in a deep-first manner in all the descending domains. If the request is denied, next possible domain should be chosen. If an indeterminate result is returned, this request is considered as permitted. It should be saved and reevaluated later. When LO is reached during the policy evaluation, LO-variable is bound with it and a planning process can be continued. The descending policy evaluation is effective, when partial policy requests are denied in high-level domains frequently, as in this case large sets of LOs can be pruned without extra policy evaluation.

6.2 Domain Refinement

An efficiency of planning depends on the possibility to make critical decisions earlier and postpone other decisions until a good opportunity occurs. The descending policy evaluation procedure was split into several stages; each stage corresponds to the policy evaluation in one domain. So decisions about execution of one stage can be taken in a flexible manner. These stages of the descending policy evaluation are realised by a domain refinement operation. In order to implement it the original planning environment is modified. All tasks are specified as $t_h := \langle T_h, L_{x_h}^{y_h} \rangle$, where T_h - an original task, $L_{x_h}^{y_h} \in L$ - a domain variable introduced. The domain variable indicates that for task T_h and for all tasks, that it'll be decomposed into, all LOs, that these tasks will be applied to, should be within the domain $L_{x_h}^{y_h} : \forall LO_i, T'_h(..LO_i..) \rightarrow LO_i \in L_{x_h}^{y_h}$. Initially all tasks correspond to a root domain. Task expansion and constraint enforcement preserve domain variables constant. In order to realise the domain refinement operation $\mathcal{R}_D(t_o)$ a special set of methods was developed. These methods substitute domain variable of the task t_o with some of its descendant domains. Alternative domain variable values form a set of alternative successor tasks $\{t_1, \dots, t_k\}$: $\mathcal{R}_D(t_o) = \{t_1, \dots, t_k\}, \forall t_i : T_i = T_o, L_{x_i}^{y_i} \subseteq L_{x_o}^{y_o}$.

The domain refinement methods are specified using a standard methods formalism, but they are applied to tasks independently from the task expansion methods (the planner chooses, which set of methods to apply). In other case, search won't be systematic: solutions for different branches won't be disjoint. Like other refinement operations (task expansion and constraint enforcement), the domain refinement operation should be sound, complete and systematic. Domain refinement completeness means that all LOs, that are descendant from original domain $L_{x_h}^{y_h}$ and that can be used in a solution, can be reached by the domain refinement. Soundness means that no other LOs can be reached. Systematic manner is achieved in case every LO is considered in one branch only. The process of task state transition is specified as an automata at figure 2. $N_{Sl}(T)$ is a number of slots corresponding to the task T . $Dom(T)$ indicates, whether the PT constraint was enforced for this task. For each combination of these parameters a special type of domain refinement methods was developed. These methods are represented using one arrow lines. Two arrows lines represent other refinements (they lead to a number of slots decreasing).

Function $PT(T) = 2^L$ is used to determine a set of PT constraints, corresponding to each compound and primitive task. In case it returns only

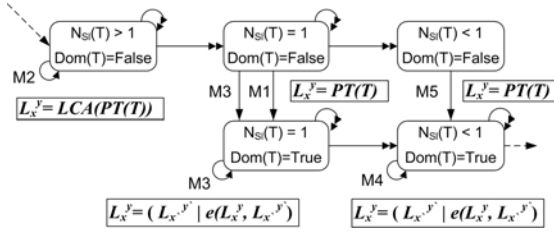


Fig. 2. State transition chart for tasks during the planning

accurate (non-probabilistic) values, there should be at most one domain refinement method applicable to some task at one time. When a domain refinement method is applied, its constraints should be enforced immediately, as they do realise the domain refinement mechanism. The domain refinement methods are applicable to a task, if it's in the state, specified at the figure, and if the current domain isn't equal to the domain, which it can be refined to. Methods M1 and M5 apply PT constraint to the task (step 1 of descending policy evaluation). Methods M3 and M4 refine domain to one of its child nodes (step 2 in descending policy evaluation). M2 refines the domain to the lowest common ancestor ($LCA(PT(T))$) for a set of domains, that are the PT constraints for this task (step 1 for a task corresponding to several slots). Methods M1 and M3 are applicable to one state, as the PT constraint for a task can become true, when method M2 is applied to this task ($Dom(T)$ flag isn't activated in this case). Methods M5 and M6 are designed in a way that all tasks, corresponding to one slot, are refined consistently. Every compound and primitive task should have all types of domain refinement methods (that could be applied to it).

6.3 Policy-Based HTN Planning Algorithm

An algorithm for policy-based HTN planning with the domain refinement described in this section is adapted from the refinement search algorithm for HTN planning in [2]. There are three types of refinements, that can be applied to a planner's search space node N : $R_E(t_i)$ - task expansion, $R_D(t_i)$ - domain refinement and $R_C(C_j)$ - constraint enforcement. Node N contains a task network TN and a set of constraints C . The choice of refinement option (one possible refinement of N) is realised using a minimum branching factor principle, according to the FAF strategy [9]. Refinement procedure at figure 3 is invoked for one node in the planner's search space. It produces a set of successor nodes and puts them into OPENLIST structure, containing nodes that are to be explored. After this a policy evaluation procedure is invoked for one node in OPENLIST. If this node is pruned during the policy evaluation, next node should be chosen.

Branching factor $K_{BE}(t_i)$ is a number of decomposition methods applicable to the task t_i , $K_{BC}(C_j)$ is a number of possible ways to enforce the constraint C_j , $K_{BD}(t_i)$ is a number of domains produced when the domain of t_i is refined. If a compound task or a constraint can't be refined further, this node should be

Refinement procedure:

1. If all tasks in N are primitive, $C = \emptyset$ and all domain variables $L_{xj}^{yj} = LO_i$, TN is solution. Return TN .
2. Compute branching factors for all refinement options in N .
3. If $K_{BE}(t_i) = K_{BD}(t_i) = 0$ for some task t_i or some $K_{BC}(C_i) = 0$, return failure.
4. Apply a refinement option with a minimum branching factor K_B .
5. Put all the nodes, generated by the refinement option, into OPENLIST. Call the policy evaluation procedure.

Policy evaluation procedure:

1. Pick one node N , that is to be evaluated, from OPENLIST.
2. In case a domain variable was substituted by a LO during the last refinement of N , check the constraints C_{LO} for LO . If constraints C_{LO} aren't satisfied, return failure.
3. Evaluate policy requests for all new tasks, generated during execution of $R_E(t_i)$ and $R_D(t_i)$, or all tasks containing variables that have been affected by $R_C(C_j)$.
 - 3.1. In case the task is primitive, generate policy request for the corresponding action.
 - 3.2. In case the task is compound, generate policy requests for all actions in a NAS for this task.
 - 3.3. Evaluate all policy requests generated during steps 3.1 - 3.2. If one policy request returns "Deny", return failure. If policy request returns "Indeterminate", remember this policy request and evaluate it when more information is accessible.
4. Call the refinement procedure for node N .

Fig. 3. Algorithm of policy-based HTN planning with domain refinement

pruned as it's a dead-end. As LO should satisfy a set of constraints $C_{LO} \subset C$ in addition to policies, in the policy evaluation procedure it should be checked that LO satisfies C_{LO} . A compound task is evaluated using its *Necessary Actions Set* (NAS), e.g. a set of actions that should be executed in every decomposition of this task.

7 Evaluation of Domain Refinement Technique

The domain refinement operation was implemented in JSHOP2, a java-based HTN planner [5]. In a testing problem domain tree is binary, it has 4 levels. LOs don't have properties, they're described by their IDs only. As descending policy evaluation was designed for large hierarchical environments with rigid constraints, we've modelled policy decisions using policy matrixes populated randomly. If a random number r is less than constant P (policy stringency measure) decision is "deny", otherwise it's "permit". So P indicates a probability to get a deny decision. There are 3 types of policies (3 policy decision matrixes): policy, that determines the possibility to apply action A_i to LO_j , policy, that determines the possibility to apply action A_i to LO_j after LO_k in a track, and policy, that determines the possibility to apply action A_i in domain L_x^y .

JSHOP2 makes decompositions in a forward manner (in the same order as they're carried out in solution) and enforces method's constraints immediately after its application. Since the planning problem considered contains only totally-ordered task networks, task, that should be refined next, is unambiguously

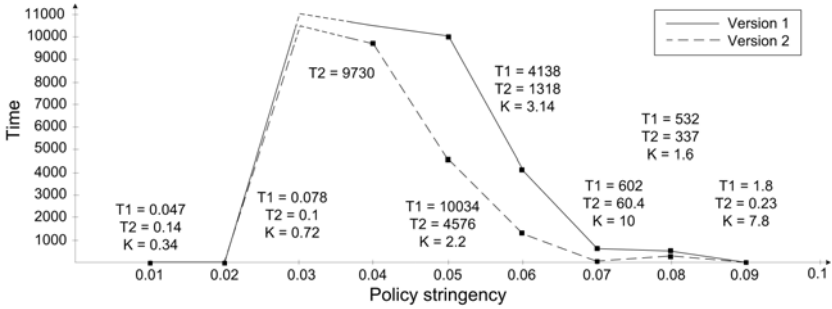


Fig. 4. Graphs of CPU time for the first problem

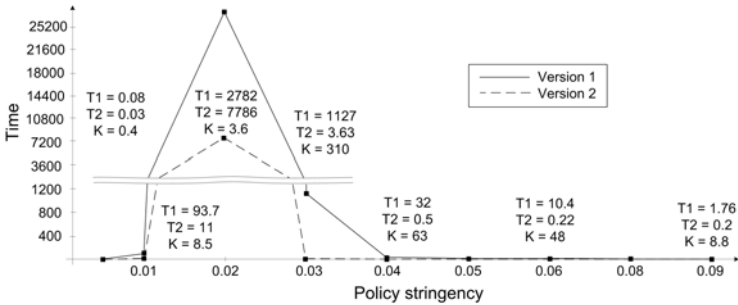


Fig. 5. Graphs of CPU time for the second problem

defined without branching factor estimation. First version of the planner uses ordinal JSHOP2 methods only. The domain refinement technique is introduced in the second version. A choice between domain refinement and task expansion is made based on branching factors. So this version implements the constrained algorithm described in section 6.3. When two planners return all possible plans for a problem, the solution sets are equal. This is an evidence of soundness and completeness of the domain refinement. Partial track for the considered planning problem is $\langle L_2^1, L_2^2, L_2^3, L_2^4, L_2^1 \rangle$, each of 8 low-level domains contains 4 LOs. Initial results indicated that version 2 of the planner finds the solution in more than 20 times faster than version 1 for all values of P . This contradicts with the theoretical assumption that the descending policy evaluation is effective when LO choice is rigidly constrained. These results were obtained as version 1 instantiates several (n) LO variables in one method. In this case JSHOP2 generates all possible LO instantiations (this leads to $n \cdot n!$ operations in the worst case). During the domain refinement each LO variable has to be instantiated using a separate method (n^n evaluations should be done in the worst case).

Further tests were carried out with a modified planning environment of version 1, where all LO variables are instantiated in separate methods. There are 40

LOs in each low-level domain (version 1 can process such task in reasonable time with exception for $P = 0.03$ and 0.04). Version 2 outperforms version 1 for bigger P (time ratio $k = t_1/t_2$ is 2 - 10 at figure 4), this confirms the assumption that the domain refinement is better for more rigid constraints. Results for the problem with partial track $\langle L_1^1, L_2^2, L_2^1, L_1^1, L_2^1 \rangle$, when even a domain for the root task can be refined, is presented at figure 5. In this case time ratio is bigger (up to 60 - 310), as domain refinement can be applied earlier during the planning. Form of the graphs depends on a partial track and on concrete policy decisions. But in general it was shown that policy refinement is better for tasks, that require more time (see figure 5), as the planner applies domain refinement more times, iterating over the same parts of the search space (as it hasn't intelligent backtracking mechanism). So the mechanism should be provided that can determine when the domain refinement technique should be applied.

8 Conclusion

In this paper the CCP generation mechanism based on HTN planning was proposed for the multi-agent e-Learning system. Limitations of HTN planning were overcome using policy-based planning approach developed. It improves the expressiveness of preconditions and facilitates specification and support of heterogeneous dynamic planning environment. In order to improve efficiency of policy-based HTN planning in large environments, new domain refinement operation was designed and implemented. Experiments've shown that domain refinement can improve the performance of policy based planning for complex problems.

References

1. AlZahrani, S., Ayesh, A., Zedan, H.: Multi-agent System Based Regional Data Grid. In: Int. Conf. on Computer Engineering & Systems, ICCES (2008)
2. Erol, K.: HTN planning: Formalization, Analysis, and Implementation. Ph.D. thesis, University of Maryland, Computer Science Department (1995)
3. Moses, T. (ed.): eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard, XACML TC (2005)
4. Nau, D., Ghallab, M., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufmann Publishers Inc., San Francisco (2004)
5. Nau, D., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: Shop2: An HTN Planning System. Journal of Artificial Intelligence Research 20, 379-404 (2003)
6. Nau, D.S., Smith, S.J., Erol, K.: Control Strategies in HTN Planning: Theory Versus Practice. In: Proceedings of AAAI 1998/IAAI 1998, pp. 1127-1133. AAAI Press, Menlo Park (1998)
7. Nikolaev, P., Ayesh, A., Zedan, H.: Combined Course Programmes Support System. In: 2nd Creativity, Innovation and Software Engineering Conference (2009)
8. Sloman, M.: Policy Driven Management for Distributed Systems. Journal of Network and Systems Management 2, 333-360 (1994)
9. Tsuneto, R., Nau, D.S., Hendler, J.A.: Plan-Refinement Strategies and Search-Space Size. In: ECP, pp. 414-426 (1997)

SimZombie: A Case-Study in Agent-Based Simulation Construction

Matthew Crossley and Martyn Amos

School of Computing, Mathematics and Digital Technology,
Manchester Metropolitan University, Manchester M15GD, UK.
{m.crossley,m.amos}@mmu.ac.uk

Abstract. In this paper we describe a general method for the conversion of an equation-based model to an agent-based simulation. We illustrate the process by converting a well-known recent case-study in epidemiology (zombie infection), and show how we may obtain qualitatively similar results, whilst gaining access to the many benefits of an agent-based implementation.

1 Introduction

Modelling is a fundamental technique for the study of a wide range of problems [5, 6, 12]. Example application domains include disease epidemics [2], crowd behaviour [24] and traffic management [20]. Many models are based on systems of *differential equations*, and examples include elastic deformation [30] and socio-economic phenomena [8]. However, such models are often difficult to *extend*, do not easily capture the notion of *space*, and fail to handle “*mixed*” populations of entities [17, 19, 27, 31]. An alternative approach, which has gained popularity in recent-years, is the *agent-based method* [18], where *individual entities* (and their interactions) within a system are explicitly *simulated* computationally [15, 21, 28]. There are many benefits to this approach, including *extendibility*, *ease of understanding* for the non-specialist, consideration of *spatial factors* and *population heterogeneity* [3, 7, 22, 31].

Although the continued rapid growth in computer power has made the agent-based approach increasingly attractive, one common barrier to its adoption has been the lack of a *coherent process* by which *existing* “traditional” models may be expressed within this *new* framework. This is the main problem that we address here. By describing in detail how an existing differential equation-based model may be *converted* into an equivalent agent-based solution, and by comparing the results obtained by each, we offer a convincing roadmap for the agent-based implementation of mathematical models. We emphasise that the problem chosen is simply used as a vehicle for illustrating the conversion process; we offer no additional insights into the fundamental nature of epidemics.

The rest of the paper is organized as follows: We briefly motivate the work, before presenting an existing epidemiological model based on “Zombie” infection, which has been successfully used as an exemplar of the equation-based approach.

We then show how the components of this model may be mapped onto an agent-based simulation framework, and compare the results obtained by each. In doing so, we describe a generic “workflow” for the conversion of such models, which is the main contribution of the paper. We conclude with a discussion of outstanding issues, and suggest future lines of enquiry.

2 Background and Motivation

Mathematical models are used to describe many different systems of interest, and facilitate analysis, prediction and optimisation [13]. Several different techniques exist for this, including statistical models [11] and game theory [25]. Another well-established modelling technique is the process of determining and subsequently evaluating a series of *ordinary differential equations* (ODEs). ODEs are used to accurately describe changes in the value of a variable, or a number of variables, with respect to time. These variables represent quantities which may change over time, such as the size of bacterial populations [4], monetary amounts [29] or plant mass [1], and there exist many numerical methods for approximating solutions to systems of ODEs [10].

Unlike ODEs, which attempt to model using a “top-down” approach (the behaviour of the system *as a whole* is described at the outset by a set of equations), an *agent-based model* (ABM) is populated with a number of autonomous *agents*, that are each assigned a set of (often very simple) rules which govern their behaviour. The microscopic actions and interactions of agents *give rise* to macroscopic, “emergent” system-level behaviour (the “bottom-up” approach) [7].

Many real world scenarios have been modelled using both the ODE approach and the agent-based approach. In these cases it is common to compare the results to determine the quantitative *and* qualitative differences between methods. This is common in fields such as ecology, using, for example, a standard predator-prey model [32]). In the molecular biology domain, a model for intracellular chemical interactions is presented by Pogson *et al.* [26] with specific reference to the *advantages* that ABM can offer. The impact of different *network structures* on both types of method is considered by Rahmandad and Sterman [27], using a common infectivity model, the Susceptible-Exposed-Infected-Removed (SIR) model.

The obvious conclusion to be drawn from this short survey is that *both* approaches to modelling have their strengths and weaknesses. This, in itself, is not a particularly helpful observation, so it is useful to then consider the *particular aspects* of each where benefits may accrue. Several papers highlight the benefits and limitations of each [17, 19, 31], which we now summarise. ODEs are often very difficult to *extend* – equations can very rapidly become intractable [26]. They do not handle *heterogenous populations* very well. They are often difficult for the *non-specialist* to understand (and thus construct), but are, however, often *quick* to compute. These disadvantages easily translate to the advantages of ABM; *additional model complexity* is relatively trivial to incorporate within ABM, fully *heterogeneous* populations are possible, and both the model design and its results may be presented in a form that is easily understood by experts in fields other

than computation or mathematics [3]. ABM is not without drawbacks, as obtaining results from such a model can often require a significant amount of computational resource. While this problem is becoming less of an issue as computational power increases and parallel architectures such as the GPU become increasingly widespread, it is important to note that ABM can take considerably longer (in terms of run time) to obtain results than a system of ODE.

The process of creating and developing an ABM has been well-documented [16], but there exists no clear guidance on how to exploit the wealth of ODE models *already available* as a potential starting point for *new*, agent-based studies. That is, it is not immediately clear how “legacy” ODE models might be converted into a more “useful”, agent-based form (in terms of being rendered extendable, implementable on parallel computers, and so on). In the following Section we describe a *general approach* for such conversions, and illustrate its application by converting a well-known epidemiological model into an ABM.

3 Methodology

In this Section we consider the model of Munz *et al.* [23], and discuss the process of converting this specific series of differential equations into an ABM. This model uses the outbreak of a zombie infection as a case-study example of an epidemic. Within the model there exist three basic classes of individual: *susceptible*, *removed* and *zombified*. These classes map closely to those of the Susceptible-Infected-Removed model; indeed, the “zombie” model is presented as an accessible introduction to the SIR approach. *Zombified* individuals attack nearby susceptible individuals in order to transmit the zombie infection. A *susceptible* individual might successfully defeat a zombified individual with some probability. In these instances, the zombie infection is fought off, and the susceptible remains unaffected, while the zombie moves to the *removed* class. Susceptible individuals may die of natural, non-zombie-related causes, entering the removed class *without* become zombified. Finally, there is a chance that members of the removed population may be “resurrected” as zombies, reflecting a natural influx of zombies from the deceased [14]. The model is described, as a series of ODE:

$$S' = \Pi - \beta SZ - \delta S \quad . \quad (1)$$

$$Z' = \beta SZ + \zeta R - \alpha SZ \quad . \quad (2)$$

$$R' = \delta S + \alpha SZ - \zeta R \quad . \quad (3)$$

where Π represents some birth rate, α represents the rate at which zombies are defeated by susceptibles, β represents the rate at which a zombie successfully transmits the “zombification” infection, δ represents the natural death rate of susceptible individuals, and ζ represents the rate at which removed individuals resurrect as zombies.

The four cornerstones of our approach are Agents, Behaviours, Parameters and Space. These components form the basis of any successful agent-based model, and we now describe in detail how each is “extracted” from the set of ODEs.

3.1 Agents

ABMs use different classes of agent to represent heterogeneous populations. Class-set extraction from an ODE system is achieved by defining each class as a *population* within the ODE model. In our example, analysis of the ODE reveals three classifications: S, R and Z (susceptible, removed and zombified, respectively).

3.2 Behaviours

These are derived from the equations themselves, unlike agents which are usually derived from the *subject* of the equations. Any transaction within an equation indicates a behaviour that must be considered. Behaviours often occur in *pairs* – in our model, for example, as the number of susceptibles shrinks due to natural deaths, the number of removed individuals grows. It is possible to identify all behaviours that affect populations from examining the ODE alone. There are two types of behaviour to note – those that occur *spontaneously* in a population, and those that occur only when a population member *interacts* with a member of either the same population or another population.

Behaviours that occur spontaneously are easily recognisable as terms which observe the form αN , where α is some parameter, and N is a population. These terms introduce a global population change; assuming the variables are positive, a negative term indicates a population decrease, and a positive term indicates a population increase. We observe such terms in two places within our example system; the terms for removing susceptible agents due to natural death (δS), and resurrecting zombified agents from removed agents due to natural infection (ζR). The δS term is identified in the removed differential and is negated in the susceptible differential, indicating the behaviour here can be phrased as “susceptible agents become removed agents at a given rate (δ).” The ζR term is positive in the zombified differential and is negated in the removed differential, indicating the behaviour can be phrased as “removed agents become zombified agents at a given rate (ζ).” In this way, we *slowly convert a set of equations into “pseudo-code”*. The second type of behaviour occurs when agents *interact*. This is, of course, a common occurrence in studies of disease, and is modelled using mass-action incidence [9]. The term for the number of susceptible agents is defined as SZ . The only remaining terms are concerned with mixing: βSZ (infection transmission) and αSZ (zombies defeated by susceptibles). The infection transmission term (βSZ) term is negated for the susceptible differential, and is positive for the zombified differential – the behaviour may therefore be phrased as “when a susceptible individual encounters a zombified individual, there is a chance (β) the susceptible becomes zombified.” The second term, for zombies defeating susceptibles (αSZ), is negated for the zombified differential and positive for the removed differential; that is, “when a susceptible individual encounters a zombified individual there is a chance (α) the zombified individual becomes removed.”

At this point, we observe that within traditional ODE systems, behaviours which do not affect population sizes are often not considered at all. This is discussed further in Section 3.4.

3.3 Parameters

Determining parameters is a case of examining the ODE system and extracting the variables. Each variable should map to a parameter that is a potential control point for the ABM. This process is often quite straightforward, particularly when performed after behaviour identification. Analysis of our differential equations yields five clear variables: birth rate (Π), zombie defeat rate (α), infection transmission rate (β), rate of death by natural causes, (δ), and the resurrection rate (ζ).

Parameters in the ODE operate slightly differently to their counterpart parameters in an ABM. Whereas rates in differential equations represent a *proportion* of the population, they apply on a per-agent basis within an ABM. We illustrate this subtle difference by taking the term δS as an example – rather than taking the population of S and multiplying it by δ to calculate the number of susceptibles that die to natural causes, we calculate this every “time step” for *each agent* in a population. For each member of the susceptible population there is a chance (δ) that the susceptible becomes removed. The net result is the same, within some acceptable stochastic variation, in that roughly the same proportion of susceptibles are removed each time step. Behaviours that occur when two agents interact are calculated in exactly the same way, although these are calculated on a collision-by-collision basis, rather than for a whole population. There also often exist some parameters that must be considered, but are not *explicitly* described within the differential equations. These parameters usually describe the initial state of the model (in this case, the initial number of susceptible, zombified and removed individuals).

3.4 Spatiality

ODEs are spatially implicit; the zombie epidemic model uses mass-action incidence to assume an evenly mixed population. ABM offer a much broader spectrum of possibilities for spatiality, and there are potentially many options for a *spatially explicit* design that may not have been previously considered. Some ABM may disregard the notion of spatiality, particularly when it is not relevant to the problem scope, but our problem relies heavily on spatial factors. The zombie infection is spread by *proximity*, and as such the locations of agents are important, as is their ability to be considered mobile. In order to accurately reflect the notion of mass-action incidence, we consider our environment to be an open expanse in which agents are free to move about as desired. Several additional parameters are required for spatial explicitness. Agents are now capable of movement, so we must now consider their *speed*. It may be appropriate, depending on the model, to assign different speeds to different populations of agents. In our example we set a speed at which susceptible agents move, and a speed at which zombified agents move. This makes it possible to set the speed of zombified agents below that of susceptible agents, in order to reflect the “shambling” gait of the “traditional” zombie, or to the same value, in order to depict faster, more contemporary zombies. Additional parameters are also required to describe the *environment* (dimensions, internal structure, and so on).

3.5 Agent-Based Design

The conversion process, as we have seen, gives rise to a natural design. ABMs are often depicted in flowchart form, so we give our final design in Figure 1.

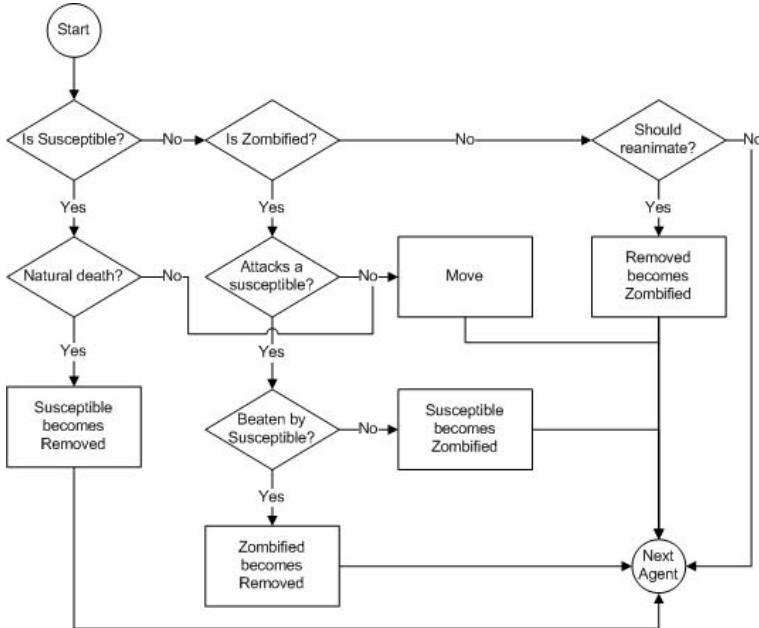


Fig. 1. Design of the agent-based model

4 Implementation and Testing

In order to test the quality of our conversion, we implemented both our ABM design and the original model [23] in a single Java application. We depict the running application in Figure 2, and the source code is available by request from the authors. The ABM results were averaged over 100 runs. In Table 1 we supply the parameter values used in both implementations (those below the bar are used only by the ABM).

We select the parameter values as follows (note that the values are the same for our implementations of both the ABM and the original model). Birth rate (Π), natural death rate (δ) and natural infection (resurrection) rate (ζ) were all set to small values (0.1%). Death rate and birth rate are described as being equal in the original model, the underlying assumption being that the “outbreak” occurs over a short period of time. These three factors are not the key focus of the model. Similarly, a high value for natural infection would model a scenario in which “zombification” occurs “naturally” rather than through transmission. We also selected a small value for “zombie defeat chance” (α) (5%), representing a population unprepared for a zombie attack. Altering this parameter would allow us to explore a

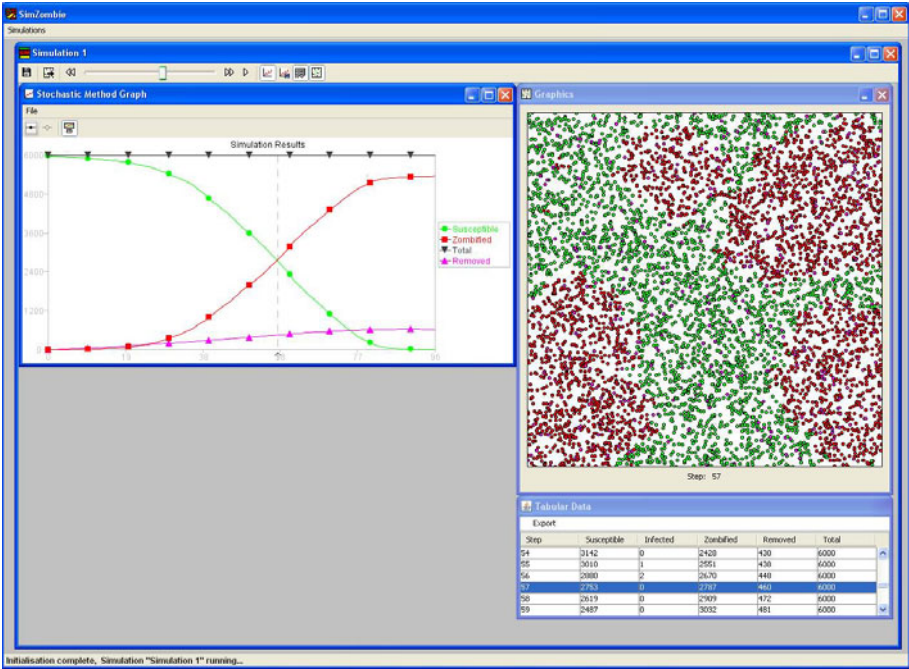


Fig. 2. Screenshot of the running application

Table 1. System parameters

Parameter	Value
Population size	6000
Birth rate, Π	0.1%
Natural death rate, δ	0.1%
Resurrection rate, ζ	0.1%
Zombie defeat chance, α	5%
Infection transmission rate, β	100%
Environment size	500 x 500 px
Infection range	10 px
Initially infected	0
Initially zombified	1
Susceptible speed	1-3 px/step
Zombie speed	1-2 px/step

scenario in which a population has some form of “resistance”. Finally, transmission rate (β) was set to 100%. This gives a scenario in which the zombie infection is always successfully transmitted. Since we are examining the spread of zombie infection, this creates a simulation with a maximum possible number of zombies due to transmission. Altering this parameter would allow us to investigate the *virulence* of infection, perhaps more closely mirroring a *realistic* scenario.

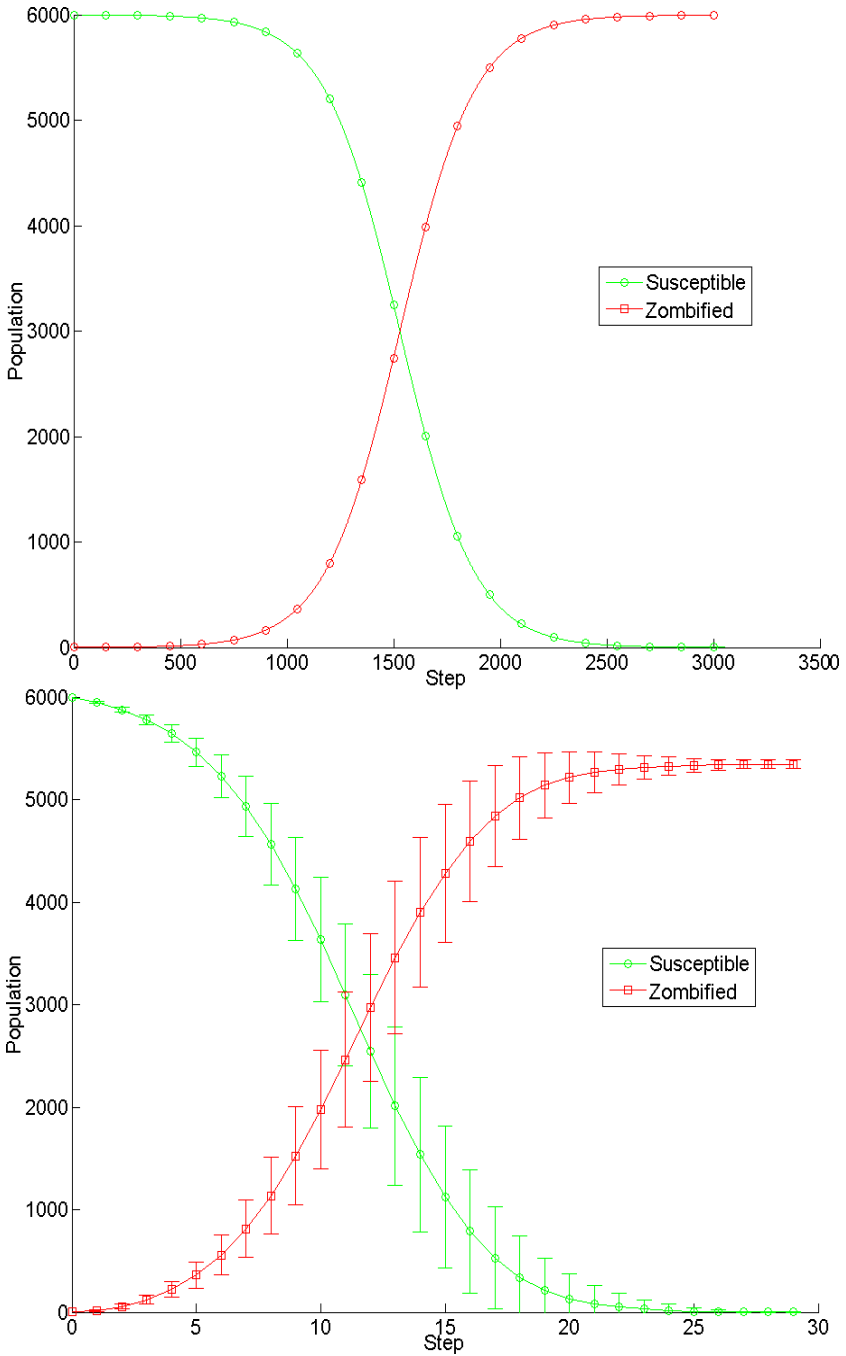


Fig. 3. Model comparison. (Top) ODE implementation; (Bottom) Agent-based implementation.

By comparing the plots obtained from both implementations (Figure 3), we observe a qualitative similarity in terms of their outputs. We therefore assert that the conversion process, from ODE to ABM, has preserved the inherent fidelity of the original model.

5 Conclusion

In this paper we describe the process of converting an ordinary differential equation model into an agent-based model with no apparent loss of accuracy. This framework unlocks a large library of models that have already been well researched and carefully verified and validated. Crucially, agent-based versions of these models will be more easily extended, and amenable to implementation on parallel architectures. Future work will use this approach to investigate a range of such models, which will, in turn, inform future refinements of the conversion process itself.

References

- [1] Aikman, D., Scaife, A.: Modelling Plant Growth Under Varying Environment Conditions in a Uniform Canopy. *Annals of Botany* 72(5), 485 (1993)
- [2] Allen, L.: Some discrete-time SI, SIR, and SIS epidemic models. *Mathematical Biosciences* 124(1), 83–105 (1994)
- [3] Axtell, R.: Why agents?: on the varied motivations for agent computing in the social sciences. *Workshop on Agent Simulation: Applications, Models and Tools* (2000)
- [4] Baranyi, J., Roberts, T., McClure, P.: A non-autonomous differential equation to model bacterial growth. *Food Microbiology* 10(1), 43–59 (1993)
- [5] Billari, F., Fent, T., Prskawetz, A., Scheffran, J.: Agent-Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences (Contributions to Economics). *Physica-Verlag* (2006)
- [6] Bolch, G.: Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. *Wiley-Blackwell* (2006)
- [7] Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America* 99(suppl. 3), 7280–7287 (2002), <http://www.pnas.org/content/99/suppl.3/7280.short>
- [8] Brajnik, G., Lines, M.: Qualitative modeling and simulation of socio-economic phenomena. *Journal of Artificial Societies and Social Simulation* 1(1) (1998), <http://www.soc.surrey.ac.uk/JASSS/1/1/2.html>
- [9] Brauer, F., Van den Driessche, P.: Models for transmission of disease with immigration of infectives. *Mathematical Biosciences* 171(2), 143–154 (2001)
- [10] Butcher, J., Corporation, E.: Numerical methods for ordinary differential equations. *Wiley Online Library, Chichester* (2008)
- [11] Draper, N., Smith, H., Pownell, E.: Applied regression analysis, vol. 706. *Wiley, New York* (1998)
- [12] Epstein, J.: Modeling civil violence: An agent-based computational approach. *Proceedings of the National Academy of Sciences of the United States of America* 99(suppl. 3), 7243 (2002)

- [13] Epstein, J.M.: Why model? *Journal of Artificial Societies and Social Simulation* 11(4), 12 (2008), <http://jasss.soc.surrey.ac.uk/11/4/12.html>
- [14] Gagne, P.: *The Zombies That Ate Pittsburgh: The Films of George A. Romero*. Dodd, Mead (1987)
- [15] Gilbert, G.: *Agent-based models*. Sage Publications, Inc., Thousand Oaks (2008)
- [16] Gilbert, N., Terna, P.: How to build and use agent-based models in social science. *Mind & Society* 1(1), 57–72 (2000)
- [17] Grimm, V.: Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling* 115(2), 129–148 (1999)
- [18] Heath, B., Hill, R., Ciarallo, F.: A survey of agent-based modeling practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation* 12(4), 9 (2009), <http://jasss.soc.surrey.ac.uk/12/4/9.html>
- [19] Huston, M., DeAngelis, D., Post, W.: New computer models unify ecological theory. *BioScience* 38(10), 682–691 (1988)
- [20] Klügl, F., Bazzan, A.L.C.: Route decision behaviour in a commuting scenario: Simple heuristics adaptation and effect of traffic forecast. *Journal of Artificial Societies and Social Simulation* 7(1) (2004), <http://jasss.soc.surrey.ac.uk/7/1/1.html>
- [21] Law, A., Kelton, W., Kelton, W.: *Simulation modeling and analysis*, vol. 2. McGraw-Hill, New York (1991)
- [22] Macy, M., Willer, R.: From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology*, 143–167 (2002)
- [23] Munz, P., Hudea, I., Imad, J., Smith?, R.: When zombies attack!: Mathematical modelling of an outbreak of zombie infection. In: Tchuente, J.M., Chiyaka, C. (eds.) *Infectious Disease Modelling Research Progress*, pp. 133–150. Nova Science Publishers, Hauppauge (2009)
- [24] Musse, S., Thalmann, D.: A model of human crowd behavior: Group inter-relationship and collision detection analysis. In: *Computer Animation and Simulation*, Citeseer, vol. 97, pp. 39–51 (1997)
- [25] Osborne, M., Rubinstein, A.: *A course in game theory*. The MIT press, Cambridge (1994)
- [26] Pogson, M., Smallwood, R., Qwarnstrom, E., Holcombe, M.: Formal agent-based modelling of intracellular chemical interactions. *Biosystems* 85(1), 37–45 (2006)
- [27] Rahmandad, H., Sterman, J.: Heterogeneity and network structure in the dynamics of diffusion: Comparing agent-based and differential equation models. *Management Science* 54(5), 998–1014 (2008)
- [28] Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34. ACM, New York (1987)
- [29] Sidrauski, M.: Rational choice and patterns of growth in a monetary economy. *The American Economic Review* 57(2), 534–544 (1967)
- [30] Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 205–214. ACM, New York (1987)
- [31] Van Dyke Parunak, H., Savit, R., Riolo, R.: Agent-based modeling vs. equation-based modeling: A case study and users guide. *Multi-Agent Systems and Agent-Based Simulation*, 10–25 (1998)
- [32] Wilson, W.: Resolving discrepancies between deterministic population models and individual-based simulations. *American Naturalist* 151(2), 116–134 (1998)

Towards Agent-Based Crowd Simulation in Airports Using Games Technology

Olivier Szymanczyk, Patrick Dickinson, and Tom Duckett

Lincoln School of Computer Science, University of Lincoln. LN6 7TS, United Kingdom
{oszymanczyk, pdickinson, tduckett}@lincoln.ac.uk

Abstract. We adapt popular video games technology for an agent-based crowd simulation in an airport terminal. To achieve this, we investigate the unique traits of airports and implement a virtual crowd by exploiting a scalable layered intelligence technique in combination with physics middleware and a social-forces approach. Our experiments show that the framework runs at interactive frame-rate and evaluate the scalability with increasing number of agents demonstrating navigation behaviour.

Keywords: multi agent modelling, pedestrian crowd simulation, game technology, game AI, physics middleware, airport simulation.

1 Introduction

The interactions, exchanges and movements of large numbers of pedestrians often lead to complex social and cultural patterns that have been studied since as early as the nineteenth century [8]. Computer scientists are using the recent rise of available computing resources to simulate such crowds. The modelling of crowds has been a significant field of research and many techniques have been developed mainly because of their utility in architecture, safety and security, computer graphics, sociology and entertainment. Despite significant progress in areas such as crowd behaviour, animation and rendering, there are still many fields in which crowd simulation requires additional research. For instance, recent video games have been criticised for feeling lifeless, and researchers have suggested to improve the playing experiences by incorporating believable virtual crowds [19].

Usually crowd simulations involve a trade-off between the complexity of each individual and the size of the crowd. Games AI developers have defined a framework in which agent intelligence is distributed throughout the terrain [21] using a spatial database approach as a solution to this trade-off. The spatial database approach was adapted into a crowd simulation application [1]. Recently games designers have started to use computationally inexpensive middleware physics engines to simulate accurate motion and reaction of objects [13]. A physics-middleware-based crowd simulation was used in our previous work [19].

This work is based on the modelling of a virtual crowd to simulate the London Gatwick Airport South Terminal using games technology. An overview layout of the airport terminal is shown in Fig. 1.



Fig. 1. London Gatwick Airport South Terminal Map. Adapted from [11]

Our aim is to design and implement a video game environment that combines a physics middleware engine [13] and an extended scalable layered intelligence technique [21], using Dijkstra's algorithm [5] to pre-compute navigation information, with social-forces driven agents [6] as characters in order to reproduce some unique and interesting airport scenarios, which would in return enrich the feelings of realism and believability for the player in a video game. Although the use of a spatial database and a physics middleware for a social-force-driven crowd simulation is not new [1, 19], to the best of our knowledge, no existing crowd simulation has specifically exploited the uniqueness of airport traveller experience or used the layered AI approach in combination with a physics middleware and the social-forces approach as a backdrop for a real-time virtual crowd setting in a video game situation. We use the available games technology, and extend them to fulfil our aims and goals.

We first discuss the unique traits of airport. Secondly we present our current framework, which is based on the spatial database approach and extend it by augmenting the spread of sociological and environmental data in each cell. This data is consequently used to compute the navigational force acting on each agent. Lastly, we evaluate the computational feasibility of our approach and we discuss its limitations and possible future work.

2 Related Work

2.1 Crowd Simulation

Several methods that face the challenges of real-time crowd simulation have been proposed [1, 7, 17, 18]. One of the earliest applications of agent-based modelling was

seen in Reynold's boid method [17], which has recently been extended onto the Sony Playstation 3 architecture to simulate up to 15,000 agents at 60 frames-per-second [18]. It was concluded that hardware and collision avoidance optimisations were required to enhance the quality of the simulation. Helbing's Social Force Model [6] describes pedestrian dynamics as a force model that takes into account all environmental and sociological forces that act upon each agent. Related work [15] implements this approach in a crowd simulation while extending the model with biologically inspired emotions. However, some undesirable features (unnatural shaking, overlapping and vibrating) were reported. A Cellular Automaton (CA) has been described as an effective technique for modelling complex emergent behaviour of crowds, which has been used to simulate evacuation systems [20]. Despite CA being described as simple and fast to implement [20], the major disadvantage is the inability of individuals to come in direct contact with other individuals. CA crowds have been criticised for providing unrealistic crowd animation [15]. Bannerjee et al. [1] uses a scalable layered intelligence technique and a Semi-Markov Decision Process for a crowd simulation. They demonstrate navigation behaviours in a hall-evacuation scenario and they evaluate the scalability and computational performance of their approach in highly crowded scenarios. Despite showing results that handle large crowds in real time, their approach does not allow agents to come into direct contact with their neighbours due to the cell size limitation. Recently, video game developers have started to use middleware physics engines [13] to simulate in-game character and object motion. A physics-middleware-based crowd simulation was implemented in [19]. Despite being able to handle large crowds at interactive frame-rate, the behaviour of agents was deemed to be too simplistic as no collision avoidance of static or dynamic entities were incorporated.

Our approach of crowd simulation combines a modified layered AI approach consisting of a physics middleware engine and Helbing's Social Forces. We extend and improve the layered AI approach and handle agent navigation using a physics middleware engine driven by the social force approach.

2.2 The Uniqueness of Airports

Airport terminals have attracted the attention of a broad range of scientists who have tried to capture and summarise the uniqueness of airport user experiences [3, 9, 10, 14, 16]. Commonly identified traits of airports are people meeting events, traveller purposes, time pressures, high levels of intrusive surveillance through security monitoring, and heightened user emotions. Each of those traits affect airport users in their behaviour and experiences, which can be said to be unique to airports.

Lee et al. [9] try to automatically track meeting events with pedestrian detection and tracking methods using airport surveillance video, due to the unique frequency and density of several different types of people meeting events. These meeting events can be, for instance, people coming towards each other, people waiting for others, a person joining an existing group, etc. They present a people trajectory tracking and meeting event method and apply it on selected video source data from London Gatwick Airport. Their results show the precision and strengths of their method, however they identified event recognition issues, such as false alarms.

Bowes [3] investigates the impact of emotion and time on the shopping behaviour of people in an international airport. The main purpose of the research was to determine if there is a relationship between an individual's emotional state as influenced by the airport terminal environment and the desire to visit an airport retail store. In their results, airport terminals retailers were compared to high-street retailers. Preliminary results identified key points which contribute to the unique nature of airports:

- *Purpose*: Shopping is not the primary purpose for visiting an airport terminal. The main purpose is to travel or meet/greet those who are travelling.
- *Time pressures*: Flight schedules require airplanes to be leaving and landing on time.
- *Heightened emotions*: Airport events (landing/departure of an airplane), time pressure and travelling purposes lead to potentially charged emotional events.

Other researchers using airport environments in their studies [10, 14, 16] also commonly noted that airports generate heightened emotions amongst their users for similar reasons as discussed above. They also examine additional qualities intrinsic to airports such as the constantly high level of security and safety measures, which result in airports being one of the most authoritarian structures available to the free use of civilians. Whilst the main purpose of the security and safety measures is to ensure a safe journey, people's privacies and emotions are altered through the various security checkpoints, the constant surveillance, the frequent safety announcements and the extra power of authorities to search and detain individuals. Such safety traits are unique to the context of airports and impact heavily on the behaviour and emotional responses of airport users.

Recently, video game research have started to exploit such unique features of airport terminals. Linehan et al. [10] developed a pervasive mobile phone game named 'BlowTooth' as a tool in order to identify whether airport safety and security measures can be used to create a novel gaming experience. A passage in the video game 'Call of Duty: Modern Warfare 2' [4] engages the player in a terrorist attack on an international airport.

Although being novel ways to use airports in entertainment there are several issues with the above approaches involving privacy and ethical implications. For example, the authors of 'BlowTooth' are aware of the edginess and risqué nature of the idea of playing pervasive games in a non-traditional playful or fun environment such as airports. The airport scene in 'Call of Duty: Modern Warfare 2' also generated discussions [7] and has been recently suggested to mirror events caused during the recent Moscow Airport attack [2], and therefore further suggesting that airports might be inappropriate environments for entertainment purposes.

3 Framework

As airports contain unique features as discussed in the previous section, we have decided to use an airport terminal, in this case the London Gatwick South Terminal [11] as a backdrop for our crowd simulation environment. To our knowledge, no

existing crowd simulation has specifically exploited the uniqueness of airport traveller experience as a backdrop for a real-time virtual crowd setting in video game situations. In this context, we set out to design and implement a video game environment that combines the middleware PhysX [13] and a scalable layered intelligence technique [21] from the video game industry, using Dijkstra's algorithm [5] to pre-compute navigation information, with social-forces [6] driven agents as characters in order to reproduce some unique and interesting airport scenarios, which would in return enrich the feelings of realism and believability for the player. The decision to use an airport scenario was chosen to demonstrate and harness its unique features and to address the lack of video game related research on real-time crowd simulation inside an airport environment.

3.1 Static and Dynamic Representation

We take as our point of departure the results generated by our previous work on virtual crowds, which simulated a large number of real-time agents in an urban context [19]. The results of the previous work demonstrate that a combination of PhysX to simulate the agents' reactive behaviour and Millington's goal oriented algorithm [12] to produce deliberative reasoning can generate real-time large crowds at interactive frame-rate (>30 frames-per-second).

In this work, we decided to use a similar hybrid architecture for simulating in real-time a large number of agents in an airport environment. We decided to combine 'high-level' deliberative reasoning (using a spatial database approach to store environmental and sociological data) and 'low-level' reactive control (using PhysX) to generate sophisticated agent behaviour. Although we only focus on basic navigational behaviour in this paper, we show that it is feasible to use our framework to populate a crowded airport environment.

The concept of spatial databases has been popularised by the video game industry [21]. The basic idea is to store any static or dynamic AI related information in several layers and have agents take decisions based on one or a combination of these layers. Data is separated into different layers, for instance, all navigation related data is stored in one layer. One major advantage of the spatial database approach is its scalability as further layers can be easily added. Banerjee et al. [1] recently extended the spatial database to simulate an agent-based crowd simulation.

We consider our environment as a 2-D surface, which is divided into cells. In contrast to the approach in [1], we do not limit ourselves to a maximum of one agent per cell. Each cell has a sufficient area to contain at least one person of average size and the total amount of individuals it can contain is limited to the cell's actual physical size. Due to the physical constraints simulated by PhysX, agents placed on the same cell do not overlap. Each cell on the layer contains an average of all the social and environmental data which affect all entities located on the same cell. On top of the grid, we have currently added four independent layers, which contain both static and dynamic data. In the experiments presented in this paper, we implemented the following layers: *physical layer*, *navigation layer*, *occupancy layer* and *area search layer*:

- *Physical layer* represents information about the environment and all the static obstacles, such as walls, stairs, lifts, info screens, etc.
- *Navigation layer* stores all path-finding data from each cell to each destination cell generated offline by the Dijkstra algorithm.
- *Occupancy layer* represents all information about the occupancy levels of each cell saving information about which individuals are currently occupying which cell.
- *Area search layer* contains information about an unblocked line-of-sight to all the cells within a limited viewing arc in front of each agent's point of view.

We view movement of agents on the 2-D surface as a sum of vectors of all separate weighted forces reflecting the different environmental or sociological influences. The method that we use is best described through the 'social-forces' model popularised by [6], which assumes that an agent's movements in a crowd is based on physical and sociological forces, which in return stimulate behaviour. Essentially, our work combines Helbing's Social Forces Model [6] with the spatial database approach as presented by [21]. We define each cell in each layer to contain relevant environmental or sociological data which is summarised into a force vector. This force is applied onto the agent to consequently generate deliberative behaviour.

For instance, in our work we define each agent A from a total of N agents ($A = A_1..A_N$) to be positioned in the virtual world W . W contains a total of M layers L , where $L=L_1..L_M$. $Layer_l(x,y)$ represents the value of cell (x,y) in layer l . An agent needs to look up all the values of its current cell in each layer, weight them using w and summarise them as a force F which is consequently applied onto the agent. The force acting on each agent at each update cycle is given by the dynamic equation:

$$F_A = \sum_{l=0}^M w_l layer_l(x,y). \quad (1)$$

If (1) is applied, the agents try to avoid static obstacles defined in the *physical layer*, while moving towards a goal cell using a pre-calculated path stored in the *navigation layer*. Simultaneously, agents try to keep a velocity-dependent distance from heavily occupied cells, identified through the *occupancy layer*, which lie within an agent's *area search layer*.

3.2 Implementation

We implemented the static and dynamic representation discussed in the previous section in Visual Studio 2008 (C++, GLUT, PhysX 2.8.4). We also implemented a simple graphical interface for evaluation purposes. The world W is defined as 200×150m in size and sub-divided into cells of 5×5m. We define the *physical layer* to a similar layout as the London Gatwick South Terminal as shown in Fig. 1. The *physical layer* is used for PhysX static object generation, off-line path generation (Dijkstra's algorithm adapted from [5]) and layout rendering. The *navigation layer* is loaded with the pre-computed path information. Fig. 2 shows an example of flow-field information stored in the *navigational layer*. The data stored in the *occupancy* and *area search* layers are dynamically updated each time an agent moves to one of

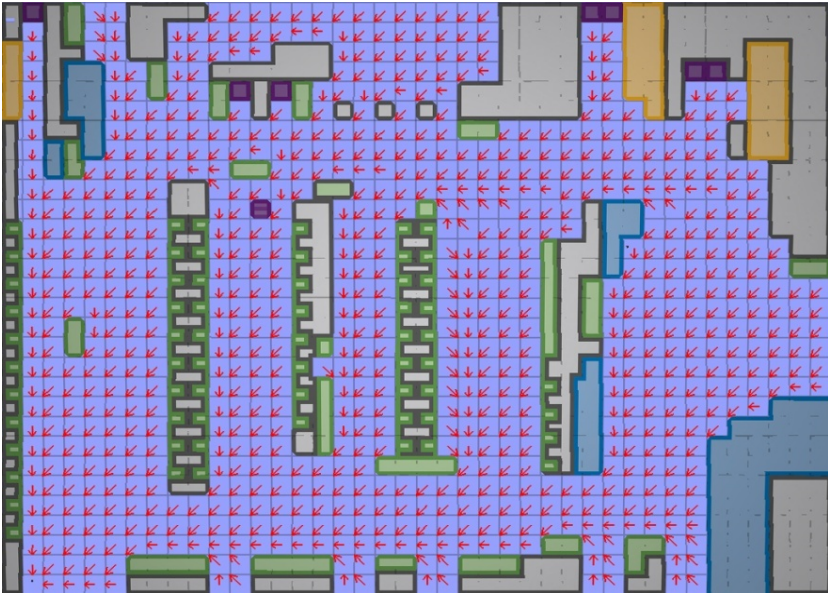


Fig. 2. Navigation flow-field produced by Dijkstra's algorithm, stored in the Navigation layer with goal the bottom-left corner



Fig. 3. Snapshot of the implementation. Agents are represented as white points, which try to follow the player controlled entity currently situated in the bottom left corner

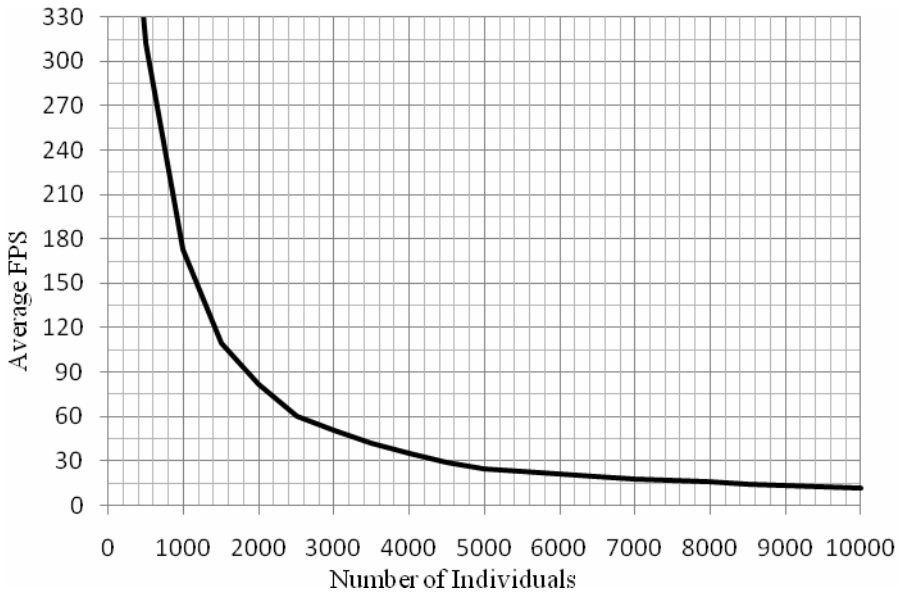


Fig. 4. Computational performance showing average frames-per-second against number of individuals in the simulation

its neighbouring cells. Agents *A* were created as dynamic PhysX actors with the size of an average person. The experiments were carried out on an Intel Core 2 Quad CPU Q6700@2.66GHz (single threaded), 2048MB Ram, GeForce 8800 GT (256MB). Fig. 3 shows a snapshot of the implementation with 1,500 agents (rendered as white dots). At the current stage of our research, the aim is to study the feasibility and scalability of our approach; therefore we recorded the frames-per-second (FPS) with population size varying between 0 and 10,000. Fig. 4 shows the recorded performance of the implementation without graphical visualisation. The FPS readings were obtained after the simulation had run for some time, allowing the average performance to be measured. The results show that the relationship between the number of individuals *N* and the performance of the application reduces exponentially with an increasing number of agents and indicates a possible bottleneck of our approach when larger populations are used.

In most video games, an acceptable minimum interactive frame-rate has been discussed to lie above thirty FPS. The system presented in this paper is able to simulate roughly up to 4,200 agents at interactive frame-rate. Whilst the current actions of agents do not follow *typical* airport behaviour, the current framework can be considered to offer sufficient performance to simulate a large crowd whilst considering future improvements.

4 Limitations and Future Work

One of the major limitations of our approach is the requirement that data contained in the *navigation layer* has to be pre-calculated offline due to its computationally

intensive requirements. This means that changes in the *physical layer*, for instance adding a new static wall section, would require the data in the *navigation layer* to be fully recalculated. Another yet unexplored limitation is the size of each cell. Currently, in our implementation, we are using cell sizes of 5×5m, however, we have not explored the impact of the granularity of the layer data and its resulting effect on the simulation if larger cells sizes are used. Currently the agent's behaviour is limited to following a player controlled entity, thus not reflecting typical airport user behaviour. In future work we plan not only to improve the current limitations but also to extend our layered AI approach to contain more layers which reflect other aspects of typical airport behaviour. We also plan to include group structures to reflect families and other small groups travelling together as a tight, closed unit. We also plan to include situational events, which in return affect an agent's current emotional status, which in return modifies the current behaviour.

5 Summary

We have presented a framework that combines a modified spatial database approach with the middleware PhysX in order to simulate a virtual crowd using the social forces approach. We have modified the spatial database approach to contain more than one agent per cell, thus augmenting the spread of sociological and environmental data in each cell. Agent overlapping is prevented by the physical boundaries generated through the middleware PhysX. Agent movement is described through the 'social-forces' approach by summarising all the relevant environmental or sociological information spread throughout the cells in each layer. Furthermore, we have applied this framework in an airport terminal environment, which has been discussed to provide a strong backdrop for a crowd simulation due to its uniqueness. We have run several experiments in order to establish the scalability and feasibility of our approach. The framework was shown to produce large crowds at interactive frame-rate. Although it was said that the current agent's behaviour did not follow a *typical* airport terminal behaviour as influenced by the unique traits of airports, the assessment of the framework's computational performances and extendibility was deemed to offer sufficient qualities, which in future work will be utilised to implement a crowd simulation that fully harnesses the uniqueness of airport terminals.

References

1. Banerjee, B., Abukmail, A., Kraemer, L.: Advancing the Layered Approach to Agent-Based Crowd Simulation. In: Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (2008)
2. BBC News: Moscow bombing Carnage at Domodedovo airport (2010), <http://www.bbc.co.uk/news/world-europe-12268662>
3. Bowes, B.: The effects of emotion and time to shop on shopping behaviour in an international airport terminal. Presentation to the Association of Consumer Research, 207–214 (2002)
4. Call of Duty Modern Warfare 2 Infinity Ward Studios Activision. Official webpage (2009), <http://modernwarfare2.infinityward.com/agegate.php>

5. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
6. Helbing, D., Molnar, P., Farkas, I., Bolay, K.: Self-organizing pedestrian movement. *Environment and Planning B* 28, 361–384 (2001)
7. Kotaku, Russian TV links Modern Warfare 2 to Russian Massacre (2010), <http://kotaku.com/5742990/russian-tv-links-modern-warfare-2-to-russian-massacre?skyline=true&s=i>
8. LeBon, G.: *Psychologie des Foules*. Alcan, Paris (1895)
9. Lee, S., Huang, C., Nevatia, R.: Definition, Detection, and Evaluation of Meeting Events in Airport Surveillance Videos. In: *TRECVID* (2008)
10. Linehan, C., Kirman, B., Lawson, S., Doughty, M.: Blowtooth pervasive gaming in unique and challenging environments. In: *Proceedings of the 28th of the International Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2695–2704 (2010)
11. London Gatwick Airport South Terminal Map. London Gatwick Airport webpage (2010), http://www.gatwickairport.com/Documents/Maps/Gatwick_South_08_10.pdf
12. Millington, I.: *Artificial Intelligence for Games*. The Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann Publishers Inc., San Francisco (2006)
13. Nvidia PhysX. PhysX webpage (2010), http://www.nvidia.com/object/physx_new.html
14. Omar, O., Kent, A.: International airport influences on impulsive shopping: trait and normative approach. *International Journal of Retail and Distribution Management* 29(5), 226–235 (2001)
15. Pelechano, N., Allbeck, J., Badler: Controlling individual agents in high-density crowd simulation. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 108–118. Eurographics Association (2007)
16. Pepe, A.: Reinventing airspace: spectatorship, fluidity, intimacy at PEK T3 ACE. *Journal of Architecture, City & Environment* 4.10, 9–19 (2009)
17. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. In: *Proceedings 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34 (1987)
18. Reynolds, C.: Big fast crowds on PS3. In: *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames* (2006)
19. Szymanczyk, O., Cielniak, G.: Emotion Modelling And Middleware Use For Group Sized Crowds In Videogames. In: *AISB 2010 Symposium: AI and Games* (2010)
20. Thalmann, D., Musse, S.R.: *Crowd Simulation*, 1st edn. Springer, Heidelberg (2007)
21. Tozour, P.: Using Spatial Database for Runtime Spatial Analysis. In: *AI Game Programming Wisdom*, vol. 2, pp. 381–390. Charles River Media, Hingham (2004)

Using Properties to Compare Both Words and Clauses

David M. Pearce, Zuhair Bandar, and David McLean

Manchester Metropolitan University, Manchester, M1 5GD, England
{D.Pearce,Z.Bandar,D.Mclean}@mmu.ac.uk

Abstract. There are many applications that use the semantic similarity of words to compare input to stored data, such as conversational agents. When a human thinks of a word they consider a meaning which has associated attributes and properties. Clauses allow words to combine together to form a single meaning. This fact allows words and clauses to be treated in the same way as a collection of properties expressed as words. In this paper, a novel mechanism is proposed to allow the similarity between two ideas expressed as properties to be found. An investigation using definitions as the source of information showed that the new method gives strong correlation.

Keywords: Conversational Agents, Word Similarity, Semantic Similarity.

1 Introduction

Similarity between ideas is extremely useful as it allows a person to make connections to be made between their existing knowledge and a new situation [1]. When people compare things they look for both similarity and dissimilarity between the items [2].

For example, a person encountering the word “osprey” for the first time would know nothing about the meaning of the word. But if they were then told that an “osprey” is a type of bird then they could retrieve the information that they know about birds and hence would know something about an osprey from this single fact. Thus they would know that an osprey has wings, a beak and would expect an osprey to be able to fly.

A previously unknown situation can be handled by using its similarity with known situations. Natural languages provide an intuitive abstraction with many ways to communicate the same idea. When an agent, such as a conversational agent, makes decisions using a natural language interface this ambiguity presents a challenge which can be overcome using semantic similarity. The associated actions or responses for a known situation are wanted to be performed when the similarity between an input and the description of a known situation is above a threshold. Words can be combined together to form more complex ideas than the words individually such as clauses and sentences. So the ability to compare words or clauses more accurately allows for superior natural language interaction.

An idea conveys a meaning that a person can break down into properties allowing a person to find the similarity between these ideas. More importantly language lets a person add details to a general idea by adding extra information such as when an adjective is associated with a noun [3] forming a clause. Words can be combined together to create more complex meanings than their individual components [4]. Clauses are examples of this where words are combined to give more detailed information. When an adjective is used it modifies or enhances the meaning of the noun with which it is associated [3].

Similarity is a powerful tool for many automated applications using language such as conversational agents [5]. There are several existing methods that are used for comparing text using similarity at clause lengths [6], [7] and longer. These consist of vector methods which combine the similarity scores for pairs of words. These systems either use a corpus for underlying knowledge, such as LSA [8] or ISIS [9] do, or use an ontological database such as WordNet [10], for instance, STASIS [11] or OMIOTIS [12]. A vector method can struggle to accurately compare a clause against a single word. The problem comes in making the association that the word Elephant is grey.

Formal ontologies such as OWL can be used to describe these relationships such as “is a” for comparing ideas [13] and WordNet contains many ontological relationships which can be described using a formal framework such as RDF [14]. Despite this ability to extend the relationships critical information such as the “is” relationship is not present so even using all of the relationships defined within WordNet [10] (such as done by OMIOTIS [12]), would give a very low similarity between the words elephant and grey. This means that a vector approach combining the word similarities would give lower similarity for the combined words “grey animal” and “elephant” than just the word “animal” and “elephant”.

This paper proposes a mechanism for comparing both words and clauses when they are expressed in terms of their key properties. There is no pre-existing database that includes the salient properties of words. So the possibility of extracting the properties from a fixed source of information for humans is also examined.

2 Method

2.1 Comparing Two words by Properties

A word can be thought of as representing an idea or thing. It is possible to express a complex idea in terms of its relationship to simpler ideas called properties that it comprises. When comparing a pair of words the similarity of the words is represented as their common properties and their dissimilarity as their distinct properties. Language can describe the individual properties as words. Figure 1 illustrates how the words “oak” and “ash” can be thought of as a set of properties. A box has been drawn to represent properties of each word for a specific meaning. In this instances both words have an “is a” relationship between themselves and tree, but it does not account for all of the similarity. For example,

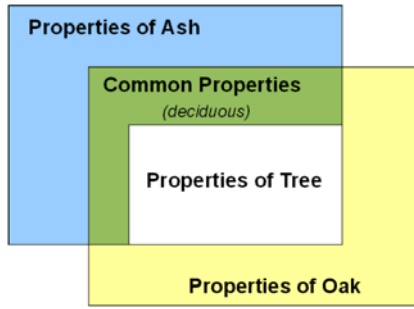


Fig. 1. Relationships between the words “oak” and “ash”

there is a concept of a deciduous tree, that adds an extra common meaning and hence a higher similarity between oak and ash.

Many WordNet based semantic similarity methods only use hypernyms to determine the similarity between a pair of words [15]. They are unable include common properties that are not represented hypernyms. Issues arise identifying the dissimilarity if all of the possible connections are used to find the similarity such as OMIOTIS [12].

If the meaning of a word is described in terms of basic properties with equal significance then the similarity is simply determined by the ratio of the common properties to all the properties (the common properties plus the distinct properties). The meaning of a word and its properties can be described using ontological relationships. The word “thing” is an example of a fundamental term because it cannot be described in simpler terms. It is equivalent to a node in a logical tree having no parent. A basic property is one that can be added directly to a fundamental term such as an adjunct. To reduce a word into these simple properties requires that some ontological relationships such as “is a” have to be followed iteratively. This is another critical difference between using properties and existing methods as it allows the specific contribution of hypernym to be added. This means that there is no need to have an algorithm with parameters to tune to a specific knowledge base which can lead to over-fitting [16].

Common properties which repeat and distinct properties could be treated as having equal weight if an absence of a property were treated as a property. So “car” compared against a “damaged car” is treated the same as if it were “undamaged car.” This lets all the basic properties be treated the same which is what is desired. However, there is one major issue that must still be addressed which is that the properties are being obtained from language. English uses a finite number of words to describe the near infinite possible meanings. This is only possible because words represent general concepts with properties that are core to their definition. However, an instance of a general concept of a word may require other properties in order to exist. These extra properties can become relevant when comparing the word and will be referred to as potential properties. This situation arises where a “red car” is being compared against a “car”. An instance of a car must have a colour and this could be “red” and so is not

definitely “not red” but a “blue car” cannot be “red”. This shows that distinct properties cannot automatically be coupled with the absence of their property.

A distinct property could be described in terms of a potential property and a value for this property. Normally the fact that a “car” has a colour is not a core defining property and so it is a potential property of the word car. In the earlier example, when the idea of a “red car” is introduced this potential property suddenly becomes relevant. Red can be thought of as being a value for the property colour.

Since each distinct property for must represent a distinct value if we take the number of distinct terms for word A as a and Word B as b then the number of distinct values scaled to a common property of magnitude 2 is simply:

$$\text{DistinctValues} = 0.5 \times (a + b). \quad (1)$$

The potential properties of a word cannot be known by its definition but can be found by considering all of the additional properties introduced by its possible hyponyms. This means that a fundamental word such as thing has a very large number of potential properties but only the properties that have a value become relevant. The potential properties can be thought of as still being present for ideas with a common hypernym, as these are not core to the definition of a word they are only relevant for distinct properties. When a potential property becomes relevant it is because one of the terms has an unknown value. This either/or situation will be treated as if the term is equally the same as different to the known value. As no nondefining property is being used for the common terms it is only the distinct potential properties that need to be considered.

To further simplify the problem it will be assumed that for every distinct value added to an idea that there was the conscious choice not to define the unknown value. This means that distinct potential properties only have to be added to the distinct properties when their values can be paired to another value, thus the distinct potential properties *DPP*:

$$DPP = \begin{cases} 0 & \text{if } a = b = 0 \\ \text{DistinctValues} \times \left(\frac{b}{a}\right) & \text{if } a \geq b \\ \text{DistinctValues} \times \left(\frac{a}{b}\right) & \text{if } b > a \end{cases}. \quad (2)$$

So the total number of distinct properties is the potential properties added to the distinct potential properties with $b \geq a$:

$$\text{Distinctproperties} = 0.5 \times (a + b) \times \left(1 + \frac{a}{b}\right). \quad (3)$$

And the similarity in terms of the common properties c is simply:

$$\text{Similarity} = \frac{c}{\left(c + 0.5 \times (a + b) \times \left(1 + \frac{a}{b}\right)\right)}. \quad (4)$$

With a scale from 1 (maximum similarity) to 0 (no common properties).

2.2 Expressing a Clause as Properties

The method used in the previous section is perhaps most powerful when it comes to its ability to be applied to clauses. The equation given in section 2.1 is only dependent upon the properties being expressed as words. So to consider the clause: “the big elephant” now it can simply be expressed as the properties of elephant combined with an extra property “big”. This information can be retrieved for future uses when the definite article acts as a specifier [4]. Any property that is combined can override an underlying property for example “a flightless bird” would have the property able to fly removed from its properties of being a bird and add the property “flightless”. The extra properties that apply to a word can be obtained from linguistic semantic relationships as shown in table 1 for the word “animal.”

Table 1. Semantic ontological relationships

Relationship	Example for animal
Adjunct	big animal
“is”	The animal “is” big
“is a”	The animal “is a” life form
“has”	The animal “has” five toes
“with”	An animal “with” a trunk

2.3 Obtaining the Properties for Words

For the word rooster an experiment was performed using the Microsoft / Encarta World English Dictionary definition [17]. This dictionary was chosen because of being available digitally and its closer relationship to an encyclopaedia and so should contain more detailed sources of information for each word.

The purpose was to determine whether the dictionary definitions would converge so as to give a number of connected words significantly smaller than the whole number of words in the vocabulary.

For each definition every noun appearing in the definition is extracted. Then each extracted nouns definition was selected for its current context and the procedure repeated.

3 Results

The total number of nouns in the iterative process for the word Rooster were 1787 (excluding undefined proper nouns) which would be less than 2.

This approach gives many unwanted properties and has also found relevant connections through a very circuitous route. Many definitions introduce association with multiple terms that contain no direct connection with one another. For example, the planet Uranus links to plays and Shakespeare because of its

moons names. Some words occur due to the commonness of the idea and the dictionary uses analogies with these ideas in order to simplify the definitions in terms that the reader should be familiar. As a result many superfluous ideas are introduced that are not connected with a rooster.

Connections between the root word “rooster” and some words connected via many steps such as molecules or the Solar system. Both of these have clear relationships with a rooster since a rooster contains molecules. However, when a person reads a sentence with the word rooster in it, it is only certain key attributes that are of interest.

3.1 Extracting Common Terms

Therefore, a much more restricted process of gathering the properties needs to be used. The following approach was adopted:

- Any common hypernym has its properties found recursively until no new hypernyms are introduced.
- When a recursive search returns to a starting term that section has its properties counted double.
- If no common hypernym is found a meronym can be searched to a depth of 2.
- A property that is given an “or” definition will score a weight of 1 if any word matches with one part.
- Adjectives, associations and combined terms each count as property with magnitude 2.

For example the Encarta [17] definition of coast is: “Land next to sea: land besides the sea.” As well as a hypernym “land” there is an association to sea for 2 but there is also a prepositional association besides.

3.2 Comparing Properties

The experiment was performed by taking the Rubenstein and Goodenough word pairs [18]. This represents the most widely used set of pairs of words and has been used by many papers. 65 pairs of nouns were compared to give human judgement as to their similarity. Table 2 gives the results of the similarity when the definitions were iteratively searched until a common term was found to a maximum of three steps from the original definition. All of the common terms are shown with the magnitude of their effect recorded. When the number is greater than 2 this means that the term is a common hypernym and if the value is 1 it means that one part of an or phrase has been matched or part of a compound clause. For example, “a lake or sea” compared against the word lake would score 1. Similarly, “with long legs” against “leg” would also score 1.

Definitions were selected so as to give the most similar definition for each word. So for example crane when compared against a cock definition as a bird. The common properties are determined by reported keywords.

Table 2. The comparison by properties of Rubenstein and Goodenoughs word pairs (RG) [18] using the Encarta dictionary [17] as a source of definitions. The total number of common properties is c , the distinct properties for word A is a and word B b and sim is the similarity using equation 4.

No.	Word A	Word B	Common Properties	c	a	b	Sim	RG
1	cord	smile		0	10	10	0.000	0.005
2	rooster	voyage		0	14	10	0.000	0.010
3	noon	string		0	8	12	0.000	0.010
4	fruit	furnace	energy(2)	2	12	14	0.077	0.013
5	autograph	shore		0	12	14	0.000	0.015
6	automobile	wizard	power(2)	2	14	12	0.077	0.028
7	mound	stove	place(1)	1	16	22	0.030	0.035
8	grin	implement		0	12	16	0.000	0.045
9	asylum	fruit	offensive term(2)	2	10	6	0.135	0.048
10	asylum	monk	community(6)	6	20	10	0.211	0.098
11	graveyard	madhouse		0	12	10	0.000	0.105
12	glass	magician		0	12	12	0.000	0.110
13	boy	rooster	living thing(6)	6	8	24	0.220	0.110
14	cushion	jewel	material(1)	1	12	12	0.040	0.113
15	monk	slave	person(10)	10	8	10	0.382	0.143
16	asylum	cemetery		0	10	8	0.000	0.198
17	coast	forest	land(14)	14	6	14	0.495	0.213
18	grin	lad	mouth(2), face(2)	4	20	8	0.169	0.220
19	shore	woodland	land(14)	14	8	8	0.467	0.225
20	monk	oracle	person(10)	10	8	10	0.382	0.228
21	boy	sage	man(12), age(1)	13	3	9	0.619	0.240
22	automobile	cushion	carry(2), things(2)	4	14	16	0.125	0.243
23	mound	shore	land(14)	14	12	8	0.457	0.243
24	wizard	lad	man(12)	12	10	2	0.625	0.248
25	forest	graveyard	place(10)	10	10	12	0.331	0.250
26	rooster	food	food(7)	7	15	0	0.483	0.273
27	cemetery	woodland	place(10)	10	10	8	0.382	0.295
28	shore	voyage	sea(2), place(2)	4	7	8	0.221	0.305
29	bird	woodland	covered(2)	2	10	14	0.089	0.310
30	coast	hill	land(14)	14	6	10	0.522	0.315
31	furnace	implement		0	12	14	0.000	0.343
32	crane	rooster	bird(14), feathers(2)	16	8	10	0.497	0.353
33	woodland	hill	land(14)	14	8	10	0.464	0.370
34	car	journey	from(2), to(2), place(2)	6	18	10	0.216	0.388
35	cemetery	mound	area(10), land(2)	12	9	8	0.428	0.423
36	glass	jewel	object(2), mineral(1)	3	14	10	0.127	0.445
37	magician	oracle	version(10)	10	6	7	0.453	0.455
38	Crane	Implement	device(22),do task(2),object(1)	25	15	10	0.545	0.593
39	Brother	Lad	man/boy(12)	12	4	2	0.727	0.603
40	Sage	Wizard	man(12), wise/clever(2)	14	6	5	0.581	0.615
41	Oracle	Sage	person(10), wisdom(2), wise(1)	13	5	5	0.565	0.653

Table 2. *Continue*

No.	Word A	Word B	Common Properties	<i>c</i>	<i>a</i>	<i>b</i>	<i>Sim</i>	RG
42	Crane	Bird	bird(14), legs(1)	15	7	0	0.811	0.658
43	Bird	Cock	bird(16)	16	0	8	0.800	0.658
44	Food	Fruit	food(6), thing(2)	8	6	7	0.399	0.673
45	Brother	Monk	man(12), religious community(2)	14	2	4	0.757	0.685
46	Asylum	Madhouse	institution(17), offensive(2), pscy- hiatric disorder(2)	21	0	3	0.933	0.753
47	Furnace	Stove	burns(2) fuel(2), heating(2), pro- duce heat(2), warm(2)	10	5	9	0.478	0.778
48	Magician	Wizard	person(10), magic(2), powers(2)	14	2	8	0.691	0.803
49	Hill	Mound	hill(26)	26	0	2	0.963	0.823
50	Cord	String	cord(20), strong(2), fibres(2), twisted(2), thin(1)	27	3	2	0.866	0.853
51	Glass	Tumbler	glass(12), no handle(1), drinking(2)	15	0	3	0.909	0.863
52	Grin	Smile	smile(30)	30	4	0	0.938	0.865
53	Serf	Slave	person(10), time(2), past(2), work(2)	16	5	4	0.664	0.865
54	Jouney	Voyage	trip(14), to(2), from(1), place(2)	19	1	4	0.859	0.895
55	Autograph	Signature	signature(20)	20	2	0	0.952	0.898
56	Coast	Shore	land(14), next(2), side(2), sea(1), water(1)	20	0	1	0.976	0.9
57	Forest	Woodland	woodland(22)	22	3	0	0.936	0.913
58	Implement	Tool	tool(24)	24	2	0	0.960	0.915
59	Cock	Rooster	cock(18)	18	2	0	0.947	0.92
60	Boy	Lad	man(12), young(2), male(2)	16	0	2	0.941	0.955
61	Cushion	Pillow	cushion(22)	22	0	5	0.898	0.96
62	Cemetery	Graveyard	ground(12), buried(2), dead(2) church(2), area(2), people(2)	22	3	5	0.775	0.97
63	Automobile	Car	vehicle(16), passenger(2), carry(2), wheels(2), engine(2), power(2)	26	0	0	1.00	0.98
64	Noon	Midday	middle of day(14), 12(2), o'clock(2), time(2)	20	0	1	0.976	0.985
65	Jewel	Gem	jewel(12), precious(2), cut(2) stone(2), polished(2)	20	0	3	0.930	0.985

The results for the 65 Rubenstein and Goodenough word pairs [18] give a strong Pearsons correlation of 0.90 and a high Spearman correlation of 0.88. In the instance of pair number 31 no connection is found between “furnace” and “implement” as a result of the definition of furnace only covering part of the potential meaning.

Several models have been evaluated against the Miller and Charles 30 word pair subset [22] of the Rubenstein and Goodenough data set. Zesch and Gurevych [16] performed a review of existing sentence similarity models using the full 65 word pairs and the best algorithms are included in table 3. They found that many of the algorithms were susceptible to over-fitting with Jiang and Conrath [23] obtaining

Table 3. Comparison with leading models

Algorithm	Pearson	Spearman
Pearce, Bandar and McLean	0.900	0.879
Leacock and Chodorow [19]	0.84 [16]	0.79 [16]
Zesch and Gurevych [20]	0.41 [16]	0.88 [16]
Alvarez and Lim [21]	0.903	

a Pearson correlation of 0.71 [16] in contrast to an earlier work by Hliaoutakis, Varelas and Voutsakis [24] which found it the best when tested on the Miller and Charles data set with a value of 0.83 [24] for the Pearson correlation. Alvarez and Lim report marginally better performance [21] for Pearson correlation but their method tunes parameters. Since equation (4) has no parameters to tune so the impact on the results from the dataset used is reduced.

4 Conclusion

This paper presents a mechanism to compare words or clauses from their properties expressed as words. It was found that, it is not all of the properties that are associated with an idea that are critical to comparing meanings but rather the properties that are semantically important to the definition of the word itself. This paper successfully shows how the properties of words represented as words themselves can be used to compare the similarity giving a strong correlation with human scores.

Probably the most powerful aspect of this result is that clauses can be treated in an identical manner to a pair of words. This means that there is the potential to extract information from text or an utterance to allow not just a comparison of a single generalised word but specific instances. Handling more complex expressions with better accuracy should allow for better performance in any system which it is integrated such as a conversational agent.

5 Further Work

The mechanism for comparing the clauses is dependent upon knowing what properties are already important to the words themselves. This produces a requirement to generate the key properties for each word and further work is needed in the automated generation of such properties.

References

1. Goldstone, R.L., Son, J.: Cambridge Handbook of Thinking and Reasoning. Cambridge University Press, Cambridge (2005)
2. Larkey, L., Markman, A.B.: Processes of similarity judgement. *Cognitive Science* 29(6), 1061–1076 (2005)

3. Taggart, C., Wines, J.A.: *My Grammar and I (or should that be 'Me'?)*. Michael O'Mara Books Limited (2008)
4. Quirk, R.: *The Use of English*, 2nd edn. Longman, Harlow (1962)
5. O'Shea, K., Bandar, Z., Crockett, K.: A Conversational Agent Framework using Semantic Analysis. *International Journal of Intelligent Computing Research* (2010)
6. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) *DaWaK 2008*. LNCS, vol. 5182, pp. 305–316. Springer, Heidelberg (2008)
7. Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1), 13–47 (2006)
8. Deerwester, S., Dumais, S.T., Landauer, G.W.F.T.K., Harshman, R.: Introduction to latent semantic analysis. *Discourse Processes* 25, 259–264 (1996)
9. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data* 2(2), 1–25 (2008)
10. Feldbaum, C. (ed.): *WordNet an Electronic Lexical Database*. MIT Press, Cambridge (1998)
11. Li, Y., Bandar, Z., McLean, D., O'Shea, J., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. *IEEE Knowledge and Data Engineering* (2006)
12. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text relatedness based on a word thesaurus. *J. Artif. Intell. Res (JAIR)* 37, 1–39 (2010)
13. Mitra, P., Wiederhold, G.: Resolving terminological heterogeneity in ontologies. In: *Proceedings of the ECAI Workshop on Ontologies and Semantic Interoperability* (2002)
14. van Assem, M., Gangemi, A., Schreiber, G.: Conversion of WordNet to a standard RDF/OWL representation. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (2006)
15. Li, Y., Bandar, Z., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Knowledge and Data Engineering* (2003)
16. Zesch, T., Gurevych, I.: Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words. *Natural Language Engineering* 16(01), 25–59 (2010)
17. *Microsoft Encarta English Dictionary*. Microsoft Corporation, Bloomsbury Publishing Plc (2004)
18. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. *Comm. ACM* 8, 627–633 (1965)
19. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, 265–283 (1998)
20. Zesch, T., Gurevych, I.: Analysis of the Wikipedia category graph for NLP applications. In: *Proceedings of the TextGraphs-2 Association for Computational Linguistics* (2007)
21. Alvarez, M.A., Lim, S.: A Graph Modeling of Semantic Similarity between Words. In: *1st IEEE International Conference on Semantic Computing*, pp. 355–362 (2007)
22. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. *Language and Cognitive Processes* 6(1), 1–28 (1991)
23. Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of Research in Computational Linguistics, Taiwan* (1998)
24. Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.: Information retrieval by semantic similarity. *Int. Journal on Semantic Web and Information Systems* 2(3), 55–73 (2006)

Semantic Similarity Measures for the Development of Thai Dialog System

Khukrit Osathanunkul, James O'Shea, Zuhair Bandar, and Keeley Crockett

Department of Computing and Mathematics, Manchester Metropolitan University,
Chester St., Manchester M1 5GD, United Kingdom
khukrit.osathanunkul@stu.mmu.ac.uk,
{j.d.oshea, z.bandar, k.crockett}@mmu.ac.uk

Abstract. Semantic similarity plays an important role in a number of applications including information extraction, information retrieval, document clustering and ontology learning. Most work has concentrated on English and other European languages. However, for the Thai language, there has been no research about word semantic similarity. This paper presents an experiment and benchmark data sets investigating the application of a WordNet-based machine measure to Thai similarity. Because there is no functioning Thai WordNet we also investigate the use of English WordNet with machine translation of Thai words.

Keywords: word-to-word similarity, word-to-word comparison, semantic similarity measures, Benchmark, Thai Dialog System.

1 Introduction

Conversational Agents (CAs) are applied in a broad range of areas including business [1], education [2] and entertainment [3]. CAs maybe used in unmanned call centres or as personal shopping or navigation assistants to reduced operating cost and provide 24/7 access for users. Most CAs use English, However some work has been in done Chinese [4] and Japanese [5]. Little or no work has been done in Thai.

The chief barrier to deploying CAs effectively in the real world is the labour cost of scripting and maintenance. Consequently, a new generation of Short Text Semantic Similarity-based agents is being introduced to overcome their problem.

Recently, a new generation of CAs has begun to emerge using Short Text Semantic Similarity measures to analyse user utterances [6]. The majority of these make use of word-to-word similarity measures based on WordNet [7]. Although work has begun on a Thai WordNet [8] it is not fully populated.

Therefore, this paper investigates the research question: Can a WordNet-based word similarity measure be developed for the Thai language by translating Thai words into English and using the English Language WordNet in a word similarity algorithm?

This work requires a Thai word similarity benchmark data set. The Thai language is the official language of Thailand, spoken by over 20 million people [9]. The Thai

alphabet uses forty-four consonants and fifteen basic vowel characters to create a word. These are horizontally placed, left to right, with no intervening space. Vowels are written after, before, below, or, above the consonant they modify, although the consonant always sounds first when the syllable is spoken.

Judgments of semantic similarity are a matter of human intuition and there is no objective method of obtaining ground truth ratings for experimental data. However, a 65 word pair data set in English collected by Rubenstein and Goodenough [10] using questionnaire-based has produced repeatable results [11].

The rest of this paper is organized as follows: Section 2 describes how the word semantic similarity measure works; Section 3 describes the collection of a Thai word similarity benchmark data set from participants using a method based on [10]; Section 4 discusses human and machine similarity ratings and section 5 outlines directions for future work.

2 The Semantic Similarity Method

There are numerous approaches to word semantic similarity; including thesaurus based [12, 13], dictionary based [14], and WordNet based [15, 16]. All use a lexical resource such as a directed graph or network and their semantic similarity is measured from the particular graph or network. WordNet [7] was developed by Princeton University, and is a machine-readable lexical database which is organized by word senses. Words in WordNet can be broken down into: 'nouns', 'verbs', 'adjectives' and 'adverbs' which are grouped into sets of synonyms. These synonyms are called 'synsets' and are connected by means of 'conceptual-semantic' and 'lexical' relations. Figure 2 is part of the hierarchy of WordNet.

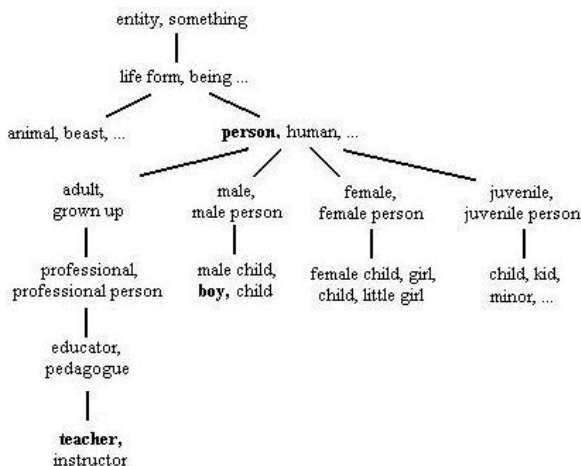


Fig. 1. Part of WordNet

WordNet approaches can be classified as: information content methods [17, 18], edge counting methods [15, 19], feature based methods [20] and hybrid methods [21]; detailed reviews of word similarity can be found in [22], [23].

In this paper, we approximate the semantic similarity between two words by using the method shown in [22] that does the estimation by looking their subsumer of two words in WordNet. According to [24], [25] the STASIS measure performs very well when compared with other measures. Also, the STASIS measure is more effective when using it automatically as it is simple and fast to calculate. The STASIS measure calculates by the following formulas:

Given two words, $w1$ and $w2$, the semantic similarity $s(w1, w2)$ (equation 1) can be calculate from:

$$s(w1, w2) = \tanh(0.45 * h) * e^{(-0.2 * d)} \quad (1)$$

Where $d1$ and $d2$ are the depth of $w1$ and $w2$, and h is the depth of their least common subsumer in WordNet, d (equation 2) can be calculate from:

$$d = d1 + d2 - (2 * h) \quad (2)$$

For example, $w1$ is *teacher* and $w2$ is *boy* in Fig. 1, the depth of $w1$ and $w2$ are 7 and 5 respectively, the synset of *person* is called the subsumer for words of *teacher* and *boy*. Therefore, h for *teacher* and *boy* is 3, and the d of *teacher* and *boy* is 6.

To address the research question in section 1, the algorithm was adapted by using machine translation of Thai words to English before submitting them to the algorithm. This was done by choosing the first sense returned by the google translation [26] utility. The Google translation engine [27] uses the United Nations parallel corpus to train their translation engine. Apart from English to Thai, it can translate over 53 languages on which further research can be done.

3 Experiment Procedures

3.1 Participants

Similarity ratings were collected from 40 native Thai speakers to create a benchmark data set. The participants had an equal number of Art/Humanities and Science/Engineering backgrounds. They consisted of 12 undergraduates and 28 postgraduates studying in 4 different UK universities. The average age of the participants was 25 and standard deviation was 2.8, with 23 male and 17 female. This is comparable with participant groups used for English word similarity [10, 11].

3.2 Materials

Following previous practice [6, 11] a representative subset of 30 word pairs was chosen from the Rubenstein and Goodenough data set. Those 30 word pairs were translated in to Thai by a native Thai speaker using the first meaning from an established Thai-English dictionary [28]. Each word pair was printed on a separate card using a standard Thai font. A questionnaire was produced containing instructions for recording similarity ratings and a small amount of personal data (Name, Confirmation of being a native Thai speaker, Age, Gender, and Academic background).

3.3 Procedure

The participants were asked to perform the following procedure:

1. Please sort the cards in to four groups in a rough order of the similarity of meaning of the word pair.
2. After sorting the cards into groups, order the cards in each group according to similarity of meaning. (i.e. the card that containing the lowest similarity of meaning is at the top of the group.)
3. Please recheck the cards in every group. You may change a pair of word to other groups at this stage.
4. Please rate the semantic similarity rating of each pair of words by writing a number between 0.0 (minimum similarity) and 0.9 for first group, 1.0 and 1.9 for second group, 2.0 to 2.9 for third group, 3.0 and 4.0 (maximum similarity) for fourth group on the recording sheet. You can use the first decimal place (e.g. 2.5) to show finer degrees of similarity. You also may assign the same value to more than one pair.

The cards were shuffled into a random order before being given to the participants.

3.4 The Benchmark Data Sets

The benchmark data set is shown in Table 1. R&G words are the original words from Rubenstein and Goodenough [10]. Translated words are the Thai words translated as described in 3.2.

Table 1. The average of similarity rating from 40 native Thai speakers

Pair Number	Word Pair				Human Rating
	Word 1		Word 2		
	R&G Word	Translated Word	R&G Word	Translated Word	
1	Cord	สายไฟ	Smile	รอยยิ้ม	0.078
5	Autograph	ลายมือชื่อ	Shore	ชายฝั่ง	0.022
9	Asylum	ที่หลบภัย	Fruit	ผลไม้	0.068
13	Boy	เด็กผู้ชาย	Rooster	นกตัวผู้	0.682
17	Coast	ฝั่งทะเล	Forest	ป่าไม้	0.632
21	Boy	เด็กผู้ชาย	Sage	นักปราชญ์	0.598
25	Forest	ป่าไม้	Graveyard	สุสาน	0.548
29	Bird	นก	Woodland	ป่าเขา	0.595
33	Hill	เนินเขา	Woodland	ป่าเขา	2.162
37	Magician	นักมายากล	Oracle	คำทำนาย	1.260
41	Oracle	คำทำนาย	Sage	นักปราชญ์	1.298
47	Furnace	เตาหลอม	Stove	เตาไฟ	1.612
48	Magician	นักมายากล	Wizard	พ่อมด	1.570
49	Hill	เนินเขา	Mound	ภูเขา	2.420
50	Cord	สายไฟ	String	เชือก	0.882
51	Glass	แก้ว	Tumbler	ถ้วยแก้ว	3.125

Table 1. (Continue)

52	Grin	ยิ้มกว้าง	Smile	รอยยิ้ม	2.330
53	Serf	ทาส	Slave	ข้ารับใช้	3.345
54	Journey	การเดินทาง	Voyage	การท่องเที่ยว	2.788
55	Autograph	ลายมือชื่อ	Signature	ลายเซ็น	3.223
56	Coast	ฝั่งทะเล	Shore	ชายฝั่ง	3.218
57	Forest	ป่าไม้	Woodland	ป่าเขา	2.830
58	Implement	อุปกรณ์	Tool	เครื่องมือ	3.335
59	Cock	ไก่ตัวผู้	Rooster	นกตัวผู้	1.515
60	Boy	เด็กผู้ชาย	Lad	เด็กหนุ่ม	2.425
61	Cushion	เบาะ	Pillow	หมอน	2.035
62	Cemetery	ป่าช้า	Graveyard	สุสาน	3.400
63	Automobile	รถยนต์	Car	รถเก๋ง	3.080
64	Midday	เที่ยงวัน	Noon	กลางวัน	3.008
65	Gem	อัญมณี	Jewel	เพชรพลอย	3.075

4 Experiment Results

4.1 Semantic Similarity Ratings

Table 2 shows the semantic similarity ratings for the translated word pairs. Column 1 is the number of the word pair as shown in table 1. Column 2 is the human rating for the Thai word pairs. Column 3 is the machine rating for the Thai word pairs using the algorithm described in section 2. Columns 4 and 5 are the human rating and the machine ratings obtain from Rubenstein and Goodenough [10]. Human ratings are calculated as the mean of the ratings provided by the set of participants for each word pair. All of the measures have been scaled in the range 0 to 1 to aid comparison.

Table 2. Semantic Similarity of Human Rating and Machine Rating

Pair Number	Thai		English	
	Human Rating	Machine Rating	Human Rating	Machine Rating
1	0.020	0.001	0.005	0.070
5	0.006	0.038	0.015	0.050
9	0.017	0.016	0.048	0.156
13	0.170	0.241	0.110	0.107
17	0.158	0.320	0.212	0.320
21	0.150	0.366	0.240	0.366
25	0.137	0.175	0.250	0.175
29	0.149	0.200	0.310	0.200
33	0.540	0.320	0.370	0.320
37	0.315	0.016	0.455	0.245
41	0.324	0.011	0.652	0.366
47	0.403	0.448	0.778	0.548
48	0.392	0.367	0.802	0.366
49	0.605	0.655	0.822	0.817
50	0.220	0.241	0.852	0.814

Table 2. (Continue)

51	0.781	0.817	0.862	0.817
52	0.582	0.667	0.865	0.667
53	0.836	0.670	0.865	0.818
54	0.697	1.000	0.895	0.547
55	0.806	1.000	0.898	0.818
56	0.804	1.000	0.900	0.817
57	0.708	1.000	0.912	1.000
58	0.834	0.547	0.915	0.817
59	0.379	0.448	0.920	1.000
60	0.606	0.670	0.955	0.670
61	0.509	0.547	0.960	0.817
62	0.850	1.000	0.970	1.000
63	0.770	1.000	0.980	1.000
64	0.752	1.000	0.985	1.000
65	0.769	1.000	0.985	1.000

4.2 Discussion

The experimental results in the previous section suggest that the Thai word-to-word semantic similarity measure and semantic similarity of human rating provides good results. As can be seen in Figure 2, most of the correlation points are near the linear line with a correlation of 0.889.

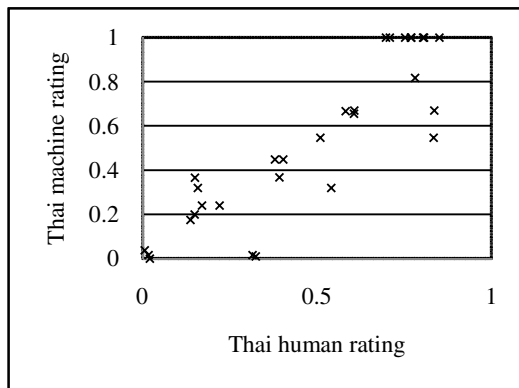


Fig. 2. The correlation between Thai human rating and Thai machine rating

Table 3 illustrates the agreement of both of the machine measures with human ratings by calculating the correlation coefficients (r) between the human ratings and the machine ratings over the data set.

Table 3. Correlation coefficients

	Correlation r
Thai human similarity rating and machine similarity measure	0.889
English human similarity rating and machine similarity measure	0.911
Thai human similarity rating and English human similarity rating	0.857
Worst Thai native speaker participant and the least of the group	0.639
Best Thai native speaker participant and the least of the group	0.946

The Thai machine measure is performing close to the English machine measure, with a different of 0.022 between the two correlation coefficients. The Thai machine measure is also performing better than the correlation between the worst performing human and the least of the group ($r = 0.639$) which supports the view that it could form the basis of an effective algorithm. Furthermore, because the best performing human achieved the correlation of 0.946 it shows this benchmark data set is capable of measuring considerable improvement over the current algorithm and should be useful to researchers on Thai semantic similarity.

Word pair 37 (Magician-Oracle) and 41 (Oracle-Sage) in Table 2 illustrate an interesting problem. Both pairs of nouns contain the word ‘oracle’. In general, ‘Oracle’ means either ‘a message given by an oracle’ or ‘someone who gave advice to people or told them what would happen’; the definition can be found in [29]. We took the first meaning from Thai-English dictionary [28] which is ‘คำทำนาย’ likely to mean ‘prediction’. After we translated the word back to English via google translate, the first meaning from google translation was chosen which is ‘prophecy’. Consequently, the machine rating that we got is low because their subsume is *entity*. The human rating was significantly higher than the machine rating as shown in table 2. This shows that the way that the machines calculate the rating for pairs of nouns is based on only this first meaning that comes up in the dictionary. On the other hand, participants reported selecting a word sense based on all of their personal knowledge of a word. The measure cannot predict which sense a human will use. Table 4 illustrates the words found to raise problems of ambiguity during machine translation.

Table 4. The exception of translate word

R&G word	Thai word	Google word	R&G word	Thai word	Google word
Cord	สายไฟ	Wire	Voyage	การท่องเที่ยว	Travel
String	เชือก	Rope	Shore	ชายฝั่ง	Coast
Sage	นักปราชญ์	Savant	Autograph	ลายมือชื่อ	Signature
Oracle	คำทำนาย	Prophecy	Jewel	เพชรพลอย	Gem
Cushion	เบาะ	Pad	Stove	เตาไฟ	Fireplace
Rooster	นกตัวผู้	Bird	Wizard	พ่อมด	Necromancer
Woodland	ป่าเขา	Forest	Implement	อุปกรณ์	Equipment
Serf	ทาส	Thrall	Asylum	ที่หลบภัย	Shadow
Automobile	รถยนต์	Car	Mound	ภูเขา	Mountain
Journey	การเดินทาง	Travel			

However, the result of this research is encouraging, and indicates the potential of semantic similarity measurement for Thai words as we explained earlier.

5 Future Work

The long term goal of the current work is to apply a word similarity measure as a component of Short Text Semantic Similarity measurement [19, 30] that can be used in Thai CAs.

We described how the Thai word-to-word semantic similarity works, showed a Thai benchmark data set, and discussed the experimental results. We are in the early stage of a completely new design to develop the Thai CAs mentioned in section 4. Further research will involve the development of the Short Text Semantic Similarity Measure for the development of a Thai dialog system. After we implement all the measures that we need to, we will develop a novel Thai CAs by using those measures.

References

1. Lemon, O., Liu, X.: DUDE: a dialogue and understanding development environment, mapping business process models to information state update dialogue systems. In: Lemon, O., Liu, X. (eds.) Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics, EACL 2006, Stroudsburg, PA, USA (2006)
2. Kopp, S., Gesellensetter, L., Krämer, N.C., Wachsmuth, I.: A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 329–343. Springer, Heidelberg (2005)
3. Ibarhim, A., Johansson, P.: Multimodal Dialogue Systems for Interactive TVApplications. In: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, ICMI 2002, Washington DC (2002)
4. Huang, F., et al.: Language understanding component for Chinese dialogue system. In: International Conference on Spoken Language Processing, Beijing, October 16-20, pp. 1053–1056 (2000)
5. Ehsani, F., Bernstein, J., Najmi, A.: An interactive dialog system for learning Japanese. Elsevier Science B.V., Amsterdam (2000)
6. O’Shea, J., Bandar, Z., Crockett, K., Mclean, D.: A Comparative Study of Two Short Text Semantic Similarity Measures. In: Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS (LNAI), vol. 4953, pp. 82–91. Springer, Heidelberg (2008)
7. Miller, G.A.: WordNet: A Lexical Database for English. *Comm. Acm* 38(11), 39–41 (1995)
8. Sornlerflamvanich, V., et al.: Review on Development of Asian WordNet. Japlo 2009 year book, 276–285 (2009)
9. Lewis, M.P. (ed.): *Ethnologue: Languages of the World*, 16th edn. SIL International, Dallas (2009)
10. Rubenstein, H., Goodenough, J.B.: Contextual Correlates of Synonymy. *Communication of the ACM* 8(10), 627–633 (1965)
11. Miller, G.A., Charles, W.G.: Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes* 6(1), 1–28 (1991)

12. Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and semantic similarity. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing, Borovetz, Bulgaria, pp. 212–219 (2003)
13. Morris, J., Hirst, J.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1), 21–48 (1991)
14. Kozima, H., Furugori, T.: Similarity between word computed by spreading activation on an English dictionary. In: Proceedings of 6th Conference of the European Chapter of the Association for Computational Linguistics, Utrecht, pp. 232–239 (1993)
15. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics* 19(1), 17–30 (1989)
16. Wu, Z., Palmer, M.: Verb semantic and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM, pp. 133–138 (1994)
17. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th international joint conference on artificial intelligence, Montreal, Canada, pp. 448–453 (1995)
18. Lin, D.: An information-theoretic definition of similarity. In: Proceeding of the 15th International Conference on Machine Learning, pp. 296–304 (1998)
19. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to Latent Semantic Analysis. *Discourse Processes* 25, 259–284 (1998)
20. Tversky, A.: Features of Similarity. *Psychological Review* 84(4), 327–352 (1977)
21. Rodriguez, M., Egenhofer, M.: Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Trans. On Knowledge and Data Engineering* 15(2), 442–456 (2003)
22. Li, Y., Bandar, Z., McLean, D.: An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Transactions on Knowledge and Data Engineering* 15(4), 871–882 (2003)
23. Pedersen, T., et al.: Measures of semantic similarity and relatedness in the Biomedical domain. *Journal of Biomedical Informatics* 40, 288–299 (2007)
24. Pirro, G.: A semantic similarity metric combining features and intrinsic information content. *Data & Knowledge Engineering* 68, 1289–1308 (2009)
25. Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.: Information retrieval by semantic similarity. *Int'l Journal on Semantic Web & Information Systems* 2(3), 55–73 (2006)
26. Google translate, <http://translate.google.com> (cited 08/10/2010)
27. Och, F.J.: *Statistical Machine Translation: Foundations and Recent Advances*. Tutorial at MT Summit 2005, Phuket, Thailand (2005)
28. Trakultaweekoon, K., Porkaew, P., Supnithi, T.: LEXiTRON Vocabulary Suggestion System with Recommendation and Vote Mechanism. In: Proceedings of Conference of SNLP 2007, Thailand (2007)
29. Longman: *Longman Dictionary of Contemporary English*, 5 edn. Longman, London (2009)
30. Li, Y., et al.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering* 18(8), 1138–1150 (2006)

Graph Coloring with a Distributed Hybrid Quantum Annealing Algorithm

Olawale Titiloye and Alan Crispin

School of Computing, Mathematics and Digital Technology
Manchester Metropolitan University
Manchester M1 5GD, U.K.

olawale.titiloye@stu.mmu.ac.uk, a.crispin@mmu.ac.uk

Abstract. Quantum simulated annealing is analogous to a population of agents cooperating to optimize a shared cost function defined as the total energy between them. A hybridization of quantum annealing with mainstream evolutionary techniques is introduced to obtain an effective solver for the graph coloring problem. The state of the art is advanced by the description of a highly scalable distributed version of the algorithm. Most practical simulated annealing algorithms require the reduction of a control parameter over time to achieve convergence. The algorithm presented is able to keep all its parameters fixed at their initial value throughout the annealing schedule, and still achieve convergence to a global optimum in reasonable time. Competitive results are obtained on challenging problems from the standard DIMACS benchmarks. Furthermore, for some of the graphs, the distributed hybrid quantum annealing algorithm finds better results than those of any known algorithm.

Keywords: Multi-agent, Optimization, Quantum Annealing, Simulated Annealing, Distributed Algorithms, Graph Coloring.

1 Introduction

Classical simulated annealing (CA) [1] is a procedure often used to find approximate solutions to difficult combinatorial optimization problems. Quantum annealing (QA) is a recent extension to simulated annealing featuring a quantum fluctuation parameter in addition to the usual temperature parameter [2,3]. QA is comparable to having multiple copies of CA executing with an interaction between them. Some earlier studies of QA addressed the travelling salesman problem [4] and Boolean satisfiability [5].

Graph coloring requires that each vertex of a given graph be assigned a color such that adjacent vertices are different, and the number of colors used is minimized. Several practical problems can be modeled as a form of the graph coloring problem. These include register allocation, radio frequency assignment, scheduling and timetabling. It is common knowledge that there is no known algorithm able to solve graph coloring exactly in the general case without requiring exponential time in the order of the problem size. Therefore there is often a need to settle for a coloring that uses a low number of colors not necessarily the minimum possible. While this approximation simplifies the problem considerably, it is still very difficult. Evolutionary algorithms which are

combined with local search are among the most powerful approaches to graph coloring [6,7,8,9].

In terms of popularity, QA is still in its infancy and some mixed results have been reported for it [3]. Even though “QA outperforms CA for most problems at least theoretically” [10]; there is a need to study individual problems independently, to check if there is sufficient reason to prefer QA. Recently, one such study was performed on graph coloring [11] using a basic algorithm similar to those in [4,5]. QA was found not only to be superior to CA, but also highly competitive with the all other graph coloring algorithms on a wide range of graphs from DIMACS [12] benchmarks. The algorithm in [11] was referred to as QA-col and a brief review of this is given in Section 2. Building on the success of the application of QA to the graph coloring, we present three further contributions which enhance the capability of QA and are likely to advance its theory and practice for optimization problems in general.

In Section 3 we introduce the idea of a fixed gamma Γ which is a control parameter in QA representing the strength of the quantum fluctuations used to perturb the system. A carefully chosen high initial value is usually set for Γ , after which it is slowly reduced towards zero during the algorithm execution. The rate of reduction of Gamma can make the difference between a successful simulation and a failed one. It had been thought that this procedure was necessary if high quality local optima were to be located in reasonable time [3,4,5]. However we empirically show that at least for graph coloring, if QA starts with a carefully chosen medium value for Γ , then no reduction needs to take place throughout the simulation. In fact this idea results in a more robust QA which locates a global minimum in a shorter time. Also, the value of a highly sensitive reduction rate no longer needs to be sought.

Section 4 addresses how mutations and population spacing ideas from evolutionary computing can be incorporated into QA to produce an algorithm that is much less likely to get trapped in local optima, and which reduces execution time considerably for many graphs. To the best of our knowledge, the hybridization of QA with mainstream evolutionary techniques is new.

We present QA algorithm modifications in Section 5 which make it possible to implement a highly scalable distributed graph coloring algorithm. Here an analogy between the modified QA and a multi-agent system is made. The resultant algorithm is referred to as DHQA (Distributed Hybrid Quantum Annealing), and it is shown by experimentation that it is very competitive with other existing graph coloring algorithms on the DIMACS [12] benchmark problems. Particularly, for some graphs, DHQA obtains better results than those found by any other algorithm in the literature. Some of the individual components of DHQA are assessed experimentally to show that they actually have a positive effect on the overall algorithm.

2 Graph Coloring and Its Basic Quantum Annealing Algorithm

What follows in this section is a brief review of the graph coloring problem and how a basic quantum annealing algorithm has been applied to it. Given a graph $G = (V, E)$, the graph coloring problem is that of coloring each vertex of G such that all adjacent vertices are colored differently with the smallest possible number of colors. This minimal number of colors is often referred to as the chromatic number or χ . Many heu-

istics including QA approach the graph coloring problem by first finding a k -coloring that uses a high number $k \leq |V|$ of colors but otherwise satisfies the constraints of the graph coloring problem. After establishing that a k -coloring exists, the same procedure is attempted for $k - 1, k - 2$ and so on until some stopping criterion such as a time limit is met. The final successful value of k is an upper bound for χ . Depending on the quality of the heuristic and the size and structure of the problem instance, the upper bound obtained can be very close or even equal to χ . In addition to the natural view of a k -coloring as an assignment problem, it can also be formulated as a partitioning of the vertex set V into k color classes such that no color class contains adjacent vertices.

If ω is an arbitrary assignment of vertices to colors, and $\omega \in \Omega$ where Ω is the search space consisting of all possible configurations, then the usual cost function or potential energy \mathcal{H}_{pot} to be minimized in order to arrive at a k -coloring can be given as $\mathcal{H}_{pot}(\omega) = C(\omega)$, where $C(\omega)$ is the number of conflicting edges. Specifically a k -coloring is found if and only if $\mathcal{H}_{pot} = 0$. An illustration of the CA approach to minimizing \mathcal{H}_{pot} can be found in [11]. Essentially the system is initialized with a single randomized configuration ω , after which a vertex still involved in conflicts is randomly chosen and its color tentatively changed to a randomly selected one. This is termed an attempted move. If the new configuration ω' has a lesser number of conflicts than before, then the new configuration is retained. Otherwise a random number $r \in [0,1)$ is generated and compared to $\exp(-\Delta\mathcal{H}_{pot}/T)$ where T is the current value of the temperature and $\Delta\mathcal{H}_{pot} = \mathcal{H}_{pot}(\omega') - \mathcal{H}_{pot}(\omega)$. If $\exp(-\Delta\mathcal{H}_{pot}/T)$ is greater than r then ω' is accepted as the new configuration but rejected otherwise. A move is said to occur any time ω' is accepted. The neighborhood of ω is the set of configurations reachable by making one move. The average size of a neighborhood can be estimated as $N = |V| \times k$. Taking M to be a fixed integer multiplier, the search alternates between carrying out $M \cdot N$ attempted moves and reducing T . The series of $M \cdot N$ moves is sometimes referred to as a Markov chain, and for CA this is equivalent to one Monte Carlo step. When using a linear schedule, the rate of decrease can be provided implicitly by specifying *MaxSteps* (a maximum number of Monte Carlo steps).

When using QA to find a k -coloring as described in [11], a population of P individuals (or replicas in QA terminology) is initialized. Each individual maintains its own configuration so that a set ϖ consisting of P separate configurations $\omega_1, \omega_2 \dots \omega_P$ always exists. In this version of QA which is also used for other problems in [4,5], each individual takes turns to carry out a procedure similar to CA. However the cost function is replaced by one that is contributed to by each individual. This is given as \mathcal{H} in (1) from [11]. The Γ -dependent term is the kinetic energy [5].

$$\mathcal{H} = \frac{1}{P} \sum_{\rho=1}^P \mathcal{H}_{pot}(\{\omega_{\rho}\}) - J_{\Gamma} \left(\sum_{\rho=1}^{P-1} \sum_i S_{i,\rho} S_{i,\rho+1} + \sum_i S_{i,1} S_{i,P} \right) \quad (1)$$

In order to define \mathcal{H} , the configuration ω_ρ of each individual ρ is expressed as a set of Boolean variables $\{S_{i,\rho}\}$. There are a number of conceivable definitions for the Boolean variables and generally there is a freedom of choice. However for graph coloring, there is a system already demonstrated to work better than others in [11] and its description follows. Each i corresponds to a unique edge $\langle u, v \rangle$ from the complete graph formed by the vertices of G . Also, there exists a Boolean variable $S_{i,\rho}$ for every such edge from the complete graph. If a Boolean variable corresponds to $i := \langle u, v \rangle$, then if vertices u and v are differently colored the variable evaluates to true. It evaluates to false otherwise. A true value is taken to be +1 and a false one is -1. This makes the function \mathcal{H} such that individuals gradually become similar to those with which they share Boolean variable products. This aids in the spreading of good building blocks between individuals.

The temperature T is kept constant in QA and the main control parameter meant to be slowly decremented is Γ , which controls the amount of artificial quantum fluctuations in the system. The value J_Γ is given as in (2), and can be described as the coupling strength between the individuals [3,5,11].

$$J_\Gamma = -\frac{T}{2} \ln \tanh\left(\frac{\Gamma}{PT}\right) > 0 \tag{2}$$

Algorithm 1. Basic quantum annealing algorithm for graph coloring

```

Input:  $G, k, P, T_0, \Gamma_0$  and  $MaxSteps$ 
Output: Coloring with lowest number of conflicts
Initialize  $T = T_0, \Gamma = \Gamma_0$  and  $\{\omega_\rho\} := \varpi$ 
While stopping condition is not met
  For  $\rho = 1, \dots, P$ 
    While  $M, N$  iterations have not been reached
      Select vertex  $v$  randomly from the list of vertices involved in conflicts in  $\omega_\rho$ 
      Move  $v$  to a new randomly selected color class, to derive  $\omega'_\rho$  and hence  $\varpi'$ 
       $\Delta\mathcal{H}_{pot} = \mathcal{H}_{pot}(\omega'_\rho) - \mathcal{H}_{pot}(\omega_\rho)$ 
       $\Delta\mathcal{H} = \mathcal{H}(\varpi') - \mathcal{H}(\varpi)$ 
      If  $\Delta\mathcal{H}_{pot} < 0$  Or  $\Delta\mathcal{H} < 0$ 
         $\varpi = \varpi'$ 
      Else
        With probability  $\exp(-\Delta\mathcal{H}/T)$ , Set  $\varpi = \varpi'$ 
    End While
  End For
   $\Gamma = \Gamma - (\Gamma_0/MaxSteps)$ 
End While

```

The pseudo-code of a basic quantum annealing algorithm for graph coloring called QA-col [11] is listed here in Algorithm 1. A Monte Carlo step is completed when each of the P individuals have completed a Markov chain. Fast incremental methods to compute $\Delta\mathcal{H}_{pot}$ and $\Delta\mathcal{H}$ are given in [11].

3 The Fixed-Gamma Quantum Annealing

The basic QA as used in [3,4,5,11] and described for graph coloring in Algorithm 1 needs three critical and highly problem instance specific parameters to be set. These are the fixed temperature T , the strength of the quantum fluctuations Γ and the rate of decrease of Γ . In Algorithm 1 we use a linear schedule which specifies a maximum number of steps *MaxSteps* that can be made during the execution of the algorithm. In this case the rate of decrease of Γ is implicitly determined. A geometric schedule could also have been used. Although it is often stated that a high value for Γ should be chosen and slowly lowered to zero, it is also recognized that Γ still has to be carefully selected, and should not be arbitrarily high if the global minimum is to be found in reasonable time [3,5]. Also, assuming a high Γ , the very slow inverse logarithmic annealing schedule associated with theorems of convergence to a global minimum is usually impractical, as this could require an infinite amount of time [10]. A low fixed temperature $T = T_q/P$ (where T_q is the temperature of the whole system [11]) usually needs to be set within a limited range that is compatible with Γ and its decrement rate. In most cases, a lot of effort goes into finding suitable values for these three parameters. This might involve trial runs on the current instance or a similar one [4,5].

It would be desirable if Γ could be constant just like T throughout the running of QA. This would mean that we only have to set two sensitive parameters rather than three, as the rate of decrease of Γ would no longer be needed. It might first appear that such a process will fail to converge. But it turns out that at least for the graph coloring, there usually exists a medium fixed Γ at which the simulation can be carried out with better results than those of the usual method of decreasing Γ . This is demonstrated empirically in Section 6. Although we believe the present work to be the first time that a version of QA with all parameters fixed is suggested and successfully implemented, there already exists a theoretical foundation for a fixed-temperature classical simulated annealing CA for some special cases based on the “finite theory of Markov chains” [14]. Also, a particular variant of CA that does not reject moves [15] was used with a fixed-temperature annealing schedule for the graph coloring algorithm in [16], which was one of the best in its time. We demonstrate empirically in Section 6 that using carefully selected fixed values for Γ and T throughout the annealing schedule still results in a convergent behavior without the burden of specifying a decrement rate.

4 Hybridization of QA with Evolutionary Techniques

One of the earliest hybrid evolutionary algorithms for graph coloring was presented in [6]. Its basic idea consisted of maintaining a population of coloring configurations, and then following a procedure of selecting two of them and applying a problem-specific crossover to create an offspring. This offspring was then improved by carrying out a Tabu local search, and then inserted back into the population by simply replacing the worst individual. This procedure was then repeated several times. Recently, two powerful new algorithms MACOL [8] and Evo-Div [9] have improved on the ideas of [6]. They both feature advanced population management or diversity

control rules. Specifically the standard set-theoretic partition distance [16] between individuals is measured and used to prevent offspring that are too similar to one or more members of the population from being introduced. This was specifically demonstrated to lead to a more robust algorithm because the population was less likely to suffer from an early loss of diversity [9]. In a study of the application of QA to satisfiability problems [5], some analogies were made between the operations of the basic QA algorithm and evolutionary algorithms. There it was recognized that both methods involve individuals of a population which evolve by taking building blocks from one another. However standard evolutionary approaches to diversity control or mutations were not actually used in their QA algorithm. This was partly because traditional evolutionary techniques such as the crossover did not seem “physically justifiable” in the context of QA. But following the success of the algorithm in [8,9], we have found a way to hybridize QA with evolutionary-based diversity control and mutations.

The standard set-theoretic partition distance $D(\rho_i, \rho_j)$ between two individuals is defined as the least number of moves required to transform ρ_i to ρ_j , and can be calculated as $|V| - S(\rho_i, \rho_j)$ where S is a similarity measure that is obtained by solving a bipartite matching problem in polynomial time as described in [8,9,16]. Individuals involved in Boolean variable products in \mathcal{H} are considered neighbors. Anytime an individual ρ has completed a Markov chain, then for its two neighbors ρ_b and ρ_f we calculate $D(\rho, \rho_b)$ and $D(\rho, \rho_f)$ respectively, and follow the recommendation of observing a minimum spacing of $R = |V|/10$ between individuals. This idea was first used in [9] and empirically proven to be effective, but this was with Tabu search. In QA, it is sufficient to maintain a distance of R only between neighbors rather than every member of the population as in [8,9], because non-neighbors usually have a low chance of becoming similar. If either of $D(\rho, \rho_b)$ and $D(\rho, \rho_f)$ are less than R then ρ is considered too close to its neighbors. If this situation occurs then ρ is forced to repeatedly search another Markov chain until a configuration is attained that observes this distance constraint. But in order to aid ρ in eventually finding a safe distance, R vertices are chosen at random and their colors are all changed randomly before the repeat search occurs. In this case, R is the mutation strength. If ρ is still not separated enough from its neighbors, the search is repeated with an increased mutation strength. This mutation mechanism is also from [9]. In contrast to [8,9], we do not attempt to replace individuals with high $C(\omega)$ by low $C(\omega)$ offspring of other individuals, even when these maintain a good distance. While this would usually be a good idea when using Tabu search, the maximization of the kinetic energy in QA could be disrupted. Note that if good values have been chosen for T and Γ then situations requiring searches to be repeated would usually be few. It is shown in Section 6 that the incorporation of these evolutionary techniques into QA makes it more robust and able to find solutions more quickly.

5 Quantum Annealing as a Distributed Algorithm

DHQA (Algorithm 2) is a distributed version of QA with fixed- Γ and evolutionary hybridization incorporated. It is described from the point of view of one agent. The

population is regarded as a multi-agent system in which each agent works to improve its own configuration in such a way that the global objective function \mathcal{H} from (1) is minimized. An agent periodically communicates its state to its neighbors by sending messages. Our main motivation for developing a distributed version is an increase in speed. Even though the serial QA in Algorithm 1 already exhibited a structure amenable to parallelization, some modifications had to be made in order to arrive at a workable distributed version with minimal synchronization overhead. As demonstrated in [11], the currently executing individual in the serial version only needs the configurations of its neighbors in order to compute $\Delta\mathcal{H}$ exactly.

Algorithm 2. Distributed Hybrid Quantum Annealing algorithm for graph coloring

Input: G, k, P, T, Γ

Output: Coloring with lowest number of conflicts

Initialize random configuration ω

Initialize random place-holder local copies ω_b and ω_f of the neighbors

While stopping condition is not met

If distances $D(\text{self}, b)$ and $D(\text{self}, f)$ are large enough (from Section 5)

Send own configuration ω to the two neighbors b and f

Receive and update ω_b and ω_f if available

Else

Mutate ω by randomly changing the color of a specified number of vertices

While $M \cdot N$ iterations have not been reached

Select vertex v randomly from the list of vertices involved in conflicts in ω

Move v to a new randomly selected color class, to derive ω'

$\Delta\mathcal{H}_{pot} = \mathcal{H}_{pot}(\omega') - \mathcal{H}_{pot}(\omega)$

Estimate $\Delta\mathcal{H}$ using ω_b and ω_f

If $\Delta\mathcal{H}_{pot} < 0$ **Or** $\Delta\mathcal{H} < 0$

$\omega = \omega'$

Else

With probability $\exp(-\Delta\mathcal{H}/T)$, **Set** $\omega = \omega'$

End While

End While

One of the goals of the distributed version is to have all agents executing concurrently and only communicating at long intervals. This will result in a situation where the states of all agents are constantly changing. A common way to cope with this situation is by tolerating the use of stale data from the neighboring individuals. This is what is done in Algorithm 2 and it is shown by the results of experiments in Section 6 that it results in no noticeable degradation in the effectiveness of the algorithm. With minimal communication between the agents this results in high scalability.

6 Experiments and Results

Experiments were performed using some of the most difficult instances from DIMACS benchmarks. Algorithm 2 (DHQA) was implemented with GNU C++, and

Table 1. Results for DHQA with 5 cores

Graph	k^*	k	T_q	Γ	a-moves	moves	t(m)	#hit
le450_25c	25	25	0.28	0.55	7.0×10^{10}	1.4×10^9	26	10/10
le450_25d	25	25	0.28	0.55	1.4×10^{11}	2.3×10^9	49	7/10
dsjc500.5	48	48	0.35	0.75	2.9×10^9	5.8×10^7	1	10/10
dsjc500.9	126	126	0.2	0.37	6.9×10^9	7.0×10^7	3	10/10
dsjc1000.1	20	20	0.44	1.05	4.7×10^9	2.3×10^8	3	10/10
dsjc1000.5	83	83	0.36	0.67	3.9×10^{10}	4.6×10^8	24	10/10
dsjc1000.9	222	222	0.19	0.375	1.8×10^{11}	8.1×10^8	84	2/10
dsjr500.5	122	122	0.15	0.1	9.6×10^9	3.1×10^8	7	5/10
r250.5	65	65	0.1	0.1	3.5×10^9	2.1×10^8	2	10/10
r1000.5	234	238	0.11	0.07	5.1×10^{10}	9.4×10^8	45	4/10
flat300_28_0	28	31	0.35	0.70	1.5×10^8	4.7×10^6	0	10/10
flat1000_76_0	82	82	0.36	0.67	4.0×10^{10}	4.7×10^8	24	10/10
latin_sqr_10	98	97	0.45	0.95	2.0×10^{11}	2.0×10^9	132	1/10
C2000.5	148	147	0.32	0.60	1.5×10^{12}	7.0×10^9	928	5/5

Our colorings are available at <http://sites.google.com/site/olawaletitiloye/graphcoloring/dhqa>

Table 2. Comparisons between DHQA and some other algorithms

Graph	k^*/χ	DH QA	[13] 1996	[17] 2008	[7] 2008	[18] 2008	[8] 2010	[9] 2010	[11] 2011
le450_25c	25/25	25	25	26	25	25	25	25	25
le450_25d	25/25	25	25	26	25	25	25	25	25
dsjc500.5	48/?	48	49	48	48	48	48	48	48
dsjc500.9	126/?	126	126	126	127	126	126	126	126
dsjc1000.1	20/?	20	21	20	20	20	20	20	20
dsjc1000.5	83/?	83	88	84	83	83	83	83	83
dsjc1000.9	222/?	222	226	224	224	225	223	223	222
dsjr500.5	122/122	122	123	125	122	125	122	122	122
r250.5	65/65	65	65	*	65	66	65	65	65
r1000.5	234/234	238	241	*	234	247	245	237	238
flat300_28_0	28/28	31	31	31	31	28	29	29	31
flat1000_76_0	82/76	82	89	84	82	87	82	82	82
latin_sqr_10	98/?	97	98	104	101	*	99	98	98
C2000.5	148/?	147	*	*	*	*	148	148	*

run on an Ubuntu Linux PC with 6GB RAM and a six-core AMD (2.6GHz each). Parallelism was achieved with the compiler’s built-in support for OpenMP (Open Multi-Processing). For each problem instance, values for T and Γ were obtained as in [11] by quick trial runs on the graph using bigger (easier) values of k . The total number of agents P was set to 20 and M the multiplier to the neighborhood size was set to 8 for all instances. Since DHQA is a distributed algorithm with agents communicating by messaging, an implementation with MPI (Message Passing Interface) would be natural. However the same effect can often be produced on a shared memory system

using OpenMP and an SPMD (Single Program Multiple Data) paradigm, which is what we used. More information on the graphs in column 1 from Table 1 can be found in [8,9]. The other columns are for the best solutions in the literature (k^*), the number of colors used by DHQA(k), T_q , Γ , the average number of attempted moves (a-moves), the average number of moves (moves), the average time t to the nearest minute, and the success ratio (#hit). Only successful runs are included in averages. The stop condition used was a time limit of 1.5 hours for all instances except latin_sqr_10 and C2000.5 which were 2.5 hours and 20 hours respectively.

In order to demonstrate the importance of the evolutionary hybridization, 10 runs were performed for the instance (dsjc1000.5, 83) with hybridization disabled and all other settings kept constant. No solutions were found. To improve the results without hybridization, Gamma had to be increased from 0.67 to 0.72 as in [11].

However, this resulted in the tripling of the execution time on the average. In Table 1, only the results of graphs dsjc500.9 and dsjc1000.9 were obtained without the hybridization component. Because of their very high density of 90%, they appear to be special cases where the current mutation idea might be counterproductive. This issue is under investigation. To estimate the speedup due to parallelization, one hundred runs of the instance (dsjc500.5, 48) were performed first with only one core and then with five cores. The average time for one core was 285 seconds, while this was 70 seconds for 5 cores. Therefore the speedup for five cores is given as 285/70 or approximately 4. This corresponds to an efficiency of about 80%. The introduction of a fixed Γ was evidently successful as the algorithm consistently achieved convergence to a global minimum in equal or better time when compared to QA-col in [11]. For the first time, it is discovered that latin_sqr_10 and C2000.5 can actually be colored with 97 and 147 colors respectively. The Latin square graph latin_sqr_10 is problematic for most algorithms, while C2000.5 is a large and difficult random graph with about one million edges. Also, only DHQA and its predecessor QA-Col find 222-colorings for the random graph dsjc1000.9. From Table 2, it can be seen that for 12 out of 14 graphs, DHQA obtains the best known result or better. None of the other algorithms achieve this score. The algorithm QA-col from [11] is the next best by this measure with 10 out of 14. Recent algorithms MACOL [8] and Evo-Div [9] both score 9 out 14. Therefore, DHQA can be considered highly competitive.

7 Conclusions

We have introduced three improvements to the basic quantum annealing for graph coloring. These consist of fixing Gamma, hybridization with evolutionary techniques and a description of a highly scalable distributed version of the algorithm. All three improvements were shown to be powerful, resulting in better solutions than any known algorithm for two well studied benchmark graphs. It is very likely that these three enhancements are transferable to other optimization problems. Instead of using only the basic QA to design algorithms for other problems, we recommend that researchers attempt to incorporate the three enhancements that we have presented.

Acknowledgements. This work is supported by the Dalton Research Institute. We thank Zuhair Bandar for discussions, and two anonymous referees for their comments.

References

1. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* 220, 671–680 (1983)
2. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. *Physical Review E* 58, 5355 (1998)
3. Das, A., Chakrabarti, B.K.: Quantum Annealing and quantum analog computation. *Rev. Mod. Phys.* 80, 1061 (2008)
4. Martoňák, R., Santoro, G.E., Tosatti, E.: Quantum annealing of the traveling salesman problem. *Phys. Rev. E* 70, 057701 (2004)
5. Battaglia, D.A., Santoro, G.E., Tosatti, E.: Optimization by quantum annealing: Lessons from hard satisfiability problems. *Phys. Rev.* 71, 066707 (2005)
6. Galinier, P., Hao, J.K.: Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization* 3(4), 379–397 (1999)
7. Malaguti, E., Monaaci, M., Toth, P.: A metaheuristic approach for the vertex coloring problem. *INFORMS Journal on Computing* 20(2), 302–316 (2008)
8. Lü, Z., Hao, J.K.: A memetic algorithm for graph coloring. *European Journal of Operational Research* 203(1), 241–250 (2010)
9. Porumbel, D.C., Hao, J.K., Kuntz, P.: An evolutionary approach with diversity guarantee and well-informed grouping recombination for graph colouring. *Computers and Operations Research* 37(1), 1822–1832 (2010)
10. Morita, S., Nishimori, H.: Mathematical foundations of quantum annealing. *Journal of Mathematical Physics* 49, 12521 (2008)
11. Titiloye, O., Crispin, A.: Quantum annealing of the graph coloring problem. *Discrete Optimization* (2011), doi:10.1016/j.disopt.2010.12.001
12. Johnson, D.S., Trick, M.: Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. DIMACS series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society, Providence (1996)
13. Morgenstern, C.: Distributed coloration neighborhood search. In: [12], pp. 335–358
14. Cohn, H., Fielding, M.: Simulated annealing: searching for an optimal temperature schedule. *SIAM Journal on Optimization* 9, 779–802 (1999)
15. Greene, J.A., Supowit, K.J.: Simulated annealing without rejected moves. *IEEE Trans. Comput.-Aided Design CAD* 5(1), 221–228 (1986)
16. Gusfield, D.: Partition-distance: a problem and class of perfect graphs arising in clustering. *Information Processing Letters* 82(3), 159–164 (2002)
17. Galinier, P., Hertz, A., Zufferey, N.: An adaptive memory algorithm for the k-coloring problem. *Discrete Applied Mathematics* 156(2), 267–279 (2008)
18. Blöchliger, I., Zufferey, N.: A graph coloring heuristic using partial solutions and a reactive tabu scheme. *Computers and Operations Research* 35(3), 960–975 (2008)

Oscar: An Intelligent Adaptive Conversational Agent Tutoring System

Annabel Latham¹, Keeley Crockett¹, David McLean¹, and Bruce Edmonds²

¹ Intelligent Systems Group, School of Computing, Mathematics & Digital Technology,
Manchester Metropolitan University, Manchester M1 5GD, UK

² Centre for Policy Modelling, Manchester Metropolitan University, Aytoun Building,
Aytoun Street, Manchester M1 3GH, UK

{A.Latham, K.Crockett, D.McLean, B.Edmonds}@mmu.ac.uk

Abstract. This paper presents an adaptive online intelligent tutoring system called Oscar which leads a tutoring conversation and dynamically predicts and adapts to a student's learning style. Oscar aims to mimic a human tutor by using knowledge of learning styles to adapt its tutoring style and improve the effectiveness of the learning experience. Learners can intuitively explore and discuss topics in natural language, helping to establish a deeper understanding of the topic and boost confidence. An initial study into the adaptation to learning styles is reported which produced encouraging results and positive test score improvements.

Keywords: Intelligent Tutoring System, Conversational Agent, Learning Style.

1 Introduction

The widespread use of the Internet has presented opportunities for the delivery of learning, both in terms of distance-learning and in supporting traditional classroom activities. Intelligent Tutoring Systems (ITS) extend traditional content-delivery computerised learning systems by adding intelligence which aims to improve the effectiveness of a learner's experience. This usually involves personalising the tutoring by adapting the learning material presented according to existing knowledge [1] or student affect such as emotion [2]. ITS which build in some social awareness, such as personalising tutoring to the individual, offer a more familiar and comfortable learning experience. Most ITS are menu-based and offer student-directed study and support at a time and pace to suit individuals, but offer an experience more akin to a computerised textbook than a classroom tutorial. Conversational Intelligent Tutoring Systems (CITS) incorporate more human-like natural language interfaces which allow learners to explore and discuss a topic, supporting the constructivist style of learning used by human tutors. However, creating a CITS which can converse naturally with a learner is a complex and time-consuming task, which is why only a few CITS exist [3][4]. Human tutors adapt their tutoring style and content based on cues they pick up

from students, such as their level of existing knowledge and their learning styles. Learning styles describe the way groups of people prefer to learn, for example by trial and error or by observation [5]. A CITS which can mimic a human tutor by leading an adaptive tutorial conversation offers students a familiar format which can help improve confidence and motivation, leading to a better learning experience. There are no tutor-led CITS which can predict and adapt to learning styles during a tutoring conversation.

This paper describes a novel CITS which dynamically predicts and adapts to a student's learning style during a tutor-led conversation. The research focussed on mimicking a face-to-face tutorial and building in knowledge of learning styles rather than designing an interface specifically to pick up learning style behaviour, as in [6]. The adaptation algorithm employed recognises the importance of providing a coherent learning experience, and so considers both the student's learning style preferences and the opportunity for adaptation in tutoring questions.

In this paper, section 2 introduces the background concepts of Intelligent Tutoring Systems, the Index of Learning Styles and Conversational Agents. Section 3 describes the Oscar CITS and the methods used to incorporate adaptivity. Section 4 outlines the experimental methodology and two sample learner dialogues. Section 5 reports the results and discussion, and Section 6 describes the conclusions and future work.

2 Background

2.1 Intelligent Tutoring Systems

Intelligent tutoring systems (ITS) are computerised learning systems which adopt intelligent systems techniques to personalise the learning experience. ITS endeavour to improve the effectiveness of tutorials and boost learners' motivation and confidence by adapting to each individual's characteristics, such as existing knowledge. ITS are normally designed to be student-directed, with a system of menu choices or hyperlinks which are reordered or ranked to recommend a particular sequence to learners [7]. Whilst this design simplifies the analysis of student behaviour, it does not truly teach the students but rather assists in self-learning, and is little different to recommending chapters of a book. Although rarely employed, conversational interfaces allow a more natural, teacher-led learning experience which supports the construction of knowledge used by human tutors [8]. Examples of CITS are AutoTutor [3] and CIRCSIM-tutor [9] which both help students construct knowledge using conversational agent tutors, however neither consider learning styles during tutoring.

The three main approaches to intelligent tutoring [1] are curriculum sequencing (presenting material in a suitable sequence [7]), intelligent solution analysis (giving feedback on incomplete or erroneous solutions [10]) and problem solving support (offering intelligent assistance in finding solutions [11]). Most ITS employ curriculum sequencing based on student knowledge and also more recently user affect factors such as emotion [12], personality [13] and learning style [4]. Few ITS incorporate all

three techniques as they are complex and time-consuming to develop, but the Oscar CITS presented in this paper will incorporate all three intelligent technologies by personalising learning material and discussing problems and solutions with students.

2.2 The Index of Learning Styles

The Index of Learning Styles (ILS) model [5] describes the learning styles in engineering education and their associated teaching styles. In the ILS model a student’s learning styles are represented as points along four dimensions to indicate the strength as well as the nature of their learning style preference. Each learning style dimension describes a step in the process of receiving and processing of information, as shown in Fig. 1. The ILS model measures learning style with a 44-question self-assessment questionnaire. There are 16 (2^4) combinations of learning styles, for example intuitive/visual/active/global.

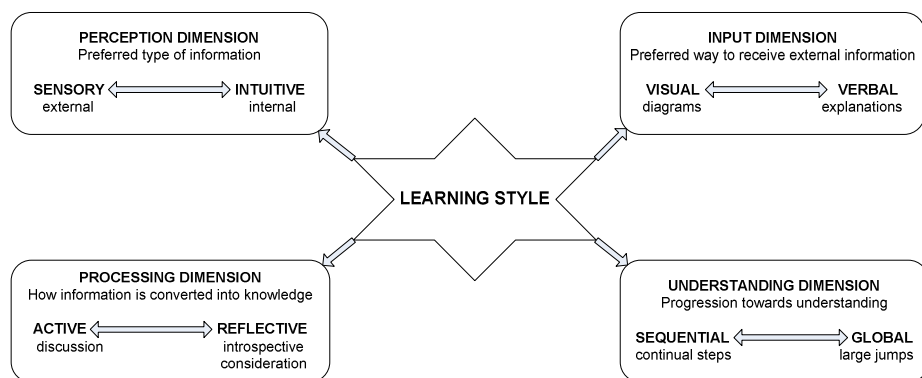


Fig. 1. ILS Dimensions

For each learning style, the ILS model details typical learner behaviours and teaching styles which address learner preferences. This information is beneficial for lecturers who informally group types of learners to adapt their teaching rather than using the formal assessment questionnaire. Knowledge of learner behaviours and teaching styles is also indispensable when developing a CITS which can adapt its teaching style to individual learner preferences.

The ILS model was incorporated into the Oscar CITS as engineering students will make up the initial experimental groups. However the flexible modular structure of the Oscar CITS does not restrict the choice of learning style model to the ILS.

2.3 Conversational Agents

Conversational agents (CAs) allow people to interact with computer systems intuitively using natural language dialogues. CA interfaces have been used effectively in many applications, such as web-based guidance [15], database interfaces [16] and

tutoring [3]. CAs are complex and time-consuming to develop, requiring expertise in the scripting of conversations, and are therefore rarely found in ITS. Systems such as Oscar CITS which aim to mimic a human tutor need CA interfaces to support the construction of knowledge through discussion [8].

Textual CAs usually adopt a pattern matching [17] or semantic based [18],[19] approach. Semantic-based CAs seek to understand the meaning of the input by studying the constructs and meanings of natural language [19] or by comparing the semantic similarity of phrases [18]. Pattern-matching CAs rely on a knowledge base containing a set of pattern-based rules [16]. During a conversation user utterances are matched to rules in the knowledge base, with the best matching rule (selected by an algorithm) firing to produce a natural language response. In the case of Oscar CITS, a pattern matching approach was adopted as it can cope with grammatically incomplete or incorrect phrases, as are commonly found in text-based chat by students.

3 Oscar: An Adaptive Conversational Intelligent Tutoring System

Oscar is an online CITS which dynamically predicts and adapts to each individual student's learning style during a tutoring conversation. By adapting the tutoring style to suit individual learners, Oscar aims to provide the most appropriate learning material for their learning style, leading to a more effective learning experience and a deeper understanding of the topic. In addition to delivering tutor material suited to an individual's learning style (known as curriculum sequencing), Oscar provides intelligent solution analysis and conversational problem solving support. Like human tutors, Oscar CITS promotes a deeper understanding of the topic by using a constructivist style of tutoring, giving intelligent hints and discussing questions with learners rather than presenting the answer straight away. Oscar CITS imitates classroom tutorials with human tutors by using a natural language interface and tutor-led tutorial style which aims to help learners feel comfortable and confident during online tutorials.

The architecture and methodology for developing the original Oscar CITS is described in [20]. Results of two initial experiments which investigated the prediction of learning styles show that Oscar CITS was successful in dynamically predicting several learning styles [20],[14]. For the initial studies, Oscar delivers an online tutorial in the domain of the database Structured Query Language (SQL). Oscar draws on knowledge bases of learning styles (the ILS model), tutor material and conversation scripts to deliver a conversational tutorial to a student. To support the tutoring conversation, diagrams, images and interactive movies may be displayed. Aspects of the student's behaviour and understanding inform the dynamic prediction of learning style, allowing the tutoring style to be personalised to best suit the student.

Throughout tutoring the Oscar CITS records and logs information about the behaviour of the student, for example the timing of interactions and the type of tutor resource accessed. The tutoring conversation is also recorded, along with information about the student knowledge of the topic being discussed.

The first implementation of Oscar CITS successfully incorporated human-like intelligence into a conversational tutorial which improved student test results and dynamically predicted their learning styles. The next section will outline the extension of Oscar CITS to include the ability to adapt a tutorial to a student's learning styles.

3.1 Methods for Including Adaptivity

The Index of Learning Styles model [5] was analysed and a table of learner behaviour for each learning style drawn up. The characteristics were evaluated to establish whether they could be incorporated into a CITS. The subset of learner behaviour considered to be most important for an adaptive CITS was then assigned the appropriate teaching styles described in the ILS model. The breakdown of behaviour and teaching styles was examined further to develop several domain-independent categories of tutor material required for developing an adaptive CITS. Each tutor material category was mapped to the appropriate learning style, for example, Category 4: Practical Examples maps to the Sensor, Active and Sequential learning styles. The standard categories were designed from the point of view of the tutor and intend to make the development of tutoring material for an adaptive CITS as simple and consistent as possible. The standard organisation of tutor material also facilitates modular development, as additional materials can be expanded and added without the need for a total redesign of the tutoring session.

The next stage was to consider how the Oscar CITS would adapt tutoring according to a student's learning style. The ILS model indicates that students who have no strong learning style preference in a dimension (i.e. they are placed at the centre of the ILS scale with a score of 1 or 3) should be given learning material including a mixture of styles. An additional *Neutral* learning style category was introduced to group those students and a Neutral adaptation style included.

There are a number of possible ways to adapt to learning styles, the simplest of which would be to adapt to the student's strongest learning style. However, a tutorial is made up of a number of tutorial questions, and this approach would require incorporating every category of tutor material into every tutorial question. This may not be possible in real life, as it is important to construct a coherent tutorial and learning experience. Consequently the adaptation strategy needed to consider not only the strength of the student's learning style but also the strength of adaptation available for each individual tutorial question. This method was adopted and a complex, domain-independent adaptation algorithm was developed which combined the strengths of the student's learning style with the tutorial adaptations to select the best fitting adaptation for each question in the student's learning path.

For the initial study an SQL revision tutorial was developed for the Oscar CITS. The adaptive SQL learning material extended the tutorial delivered in previous experiments [20],[14]. This was achieved by adding different resources covering the standard categories of tutoring material. This involved creating several versions of the learning material, each suited to a different learning style. Next, each tutorial question was assigned a score for every learning style which represented the number (or strength) of opportunities for adaptation to that learning style. Where no adaptation

existed for a learning style (i.e. the question score was zero) the Neutral adaptation was assigned by the algorithm. The initial study will now be described.

4 Experimental Methodology

A controlled study was conducted to test the hypothesis that students who are presented with learning material matched to their learning styles perform better than students presented with learning material which is unsuited to their learning styles. 70 final year undergraduate science and engineering students were asked to refresh their SQL knowledge by completing the Oscar CITS SQL revision tutorial. This involved each student registering with the Oscar CITS anonymously and completing the formal ILS questionnaire before beginning the tutorial. Next, students completed a pre-tutorial multiple choice question (MCQ) test to assess existing knowledge before starting the conversational tutorial. The tutorial was led by the Oscar CITS tutor who conversed in natural language with students and guided them through the ten tutorial questions, showing images, movies and examples as necessary. The conversational SQL revision tutorial took on average approximately 43 minutes, with each learner following an individual learning path depending on their knowledge and learning styles (see section 4.1 for example dialogues). After the tutorial conversation, students completed the same MCQ test and were then presented with a comparison of their test results and some feedback from Oscar. Finally, students were asked to complete a user evaluation questionnaire.

After completing the ILS questionnaire, participants were unknowingly assigned to one of three experimental groups. Students whose learning styles were at the centre of all ILS scales (i.e. there was no strong preference) were assigned to the *Neutral* group. These students followed the neutral adaptation learning path, with tutor material including different aspects of all learning styles (e.g. describing theory as well as examples). Students who had at least one preferred learning style were randomly assigned to either the *Adapt* or *Mismatch* groups using a 2:1 ratio. These students followed an adaptive learning path assigned by the algorithm, and were given tutor material favouring particular learning styles (e.g. containing explanations of theory rather than practical examples). Students in the Mismatch group were deliberately presented with learning material unsuited to their learning styles, e.g. if a student was Sequential, they were presented with material suited to a Global learner.

For each completed tutorial, log files which recorded the dialogue, timings, knowledge and other behavioural factors were analysed to assess each learner's performance during the tutoring session. During the SQL Revision tutoring session, ten questions were posed, requiring eighteen answers (as some questions incorporated multiple steps or questions). Learning gain was calculated by counting the number of correct answers given to tutorial questions, and an average score was calculated for each experimental group. The experimental group averages were then compared to determine whether there was any difference in learning gain related to the type of adaptation to learning style. In addition, the learner experiences in using Oscar CITS were evaluated by analysing the user feedback questionnaires.

4.1 Examples of Adaptive Learner Dialogues

Examples of the Oscar CITS tutoring style adaptation can be seen in the dialogue snippets logged during the experiment (Tables 1 and 2). Both snippets are taken from the same tutorial question where students are asked to write an SQL query, and both students were assigned to the Adapt experimental group (i.e. the adaptation matched their learning style). The student dialogue in Table 1 is an example of the tutoring style adapted to a Sequential learner, where Oscar guides the student through the steps of the query. Table 2 in contrast shows a student dialogue with tutoring adapted for a Global learner, where the student writes a query without initial guidance and learns from the feedback given about errors in their solution.

Table 1. Dialogue snippet logged during the experiment adapting to a Sequential learner

Utterance	Dialogue
87. Oscar>	Now let's see if you can remember how to write a query which chooses values between a range. Look at the employees table. We need a query which displays the first and last names of all employees hired between 1 January 1994 and 1 January 1995. The first step in writing a query is to state the information you need. For our query to display the first and last names of all employees hired between 1 January 1994 and 1 January 1995, which columns do we need?
88. Student>	dont know
89. Oscar>	Look again at the columns in the EMPLOYEES table. If we want to display the first and last names of all employees hired between 1 January 1994 and 1 January 1995, which columns do we need?
90. Student>	first_name, last_name, hire_date
91. Oscar>	That's right, we need FIRST_NAME and LAST_NAME, so the first part of our query is shown in the picture. The next step in writing a query is to state which table the information is retrieved from. For our query to list the first and last names of all employees hired between 1 January 1994 and 1 January 1995, type in the name of the table our information will be taken from.
92. Student>	employees
93. Oscar>	That's right, the table is EMPLOYEES. The last part of writing a query is the WHERE clause, which lets us restrict the rows selected to only those satisfying one or more conditions. We want to restrict the employees displayed to only those hired between 1 January 1994 and 1 January 1995. There are a number of ways to write this restriction in ORACLE SQL as dates can be represented in different formats. Type in the WHERE clause which limits the employees in this way.
94. Student>	where hire_date in('01-01-1994' && '01-01-1995')
95. Oscar>	One way to tackle this query is to use the BETWEEN keyword to specify a range. Complete the last line of the query shown - type in the WHERE clause to limit the employees displayed to those hired between 1 January 1994 and 1 January 1995.
96. Student>	'01-01-1994' and '01-01-1995'
97. Oscar>	OK, excluding date conversion functions, ORACLE recognises dates in 3 basic formats: '01-JAN-94', '01-January-94' and '01-January-1994'. The picture shows two different ways in which the WHERE clause could have been written to give the right results.

Table 2. Dialogue snippet logged during the experiment adapting to a Global learner

Utterance	Dialogue
64. Oscar>	Now let's see if you can remember how to write a query which chooses values between a range. Look at the employees table. Please type in a query which displays the first and last names of all employees hired between 1 January 1994 and 1 January 1995.
65. Student>	select First_name, last_name from Employees where Hire_date between 1-JAN-1944 and 1-Jan-1995;
66. Oscar>	I'm afraid your WHERE clause is not quite right. Check your spelling and type in the last part of our query which displays the first and last names of all employees hired between 1 January 1994 and 1 January 1995.
67. Student>	select First_name, last_name from Employees where Hire_date between 1-JAN-1994 and 1-Jan-1995;
68. Oscar>	No. Excluding date conversion functions, ORACLE recognises dates in 3 basic formats: '01-JAN-94', '01-January-94' and '01-January-1994'. The picture shows two different ways in which the WHERE clause could have been written to give the right results.

5 Results and Discussion

Of the 70 student participants, 54 fully completed the tutoring session. Table 3 shows the results of the experiment. The group of 8 Neutral students had learning style results which showed no strong preference for a particular learning style (i.e. their styles were balanced in the centre of the scale), and followed a neutral adaptation learning path. The Adapt group contained 32 students who followed a learning path containing learning material in a style adapted to their individual learning styles. The Mismatch group of 14 students followed an adaptive learning path of tutor material which was mismatched to their learning styles.

Table 3. Experimental results

Experimental Group	No. Students	Average Correct Answers
Neutral	8	72%
Adapt	32	73%
Mismatch	14	61%
Total	54	70%

Students in the Neutral and Adapt groups have similar averages of correct answers given during the tutoring, of 72% and 73% respectively. However, the Mismatch group has a much lower average of only 61% correct answers, which is 12% less than the Adapt group average. The results support the hypothesis that students who are presented with learning material matched to their learning styles perform better than students presented with learning material which is not matched to their learning styles.

In general, the user evaluation results showed that the Oscar CITS was well received, with 95% of learners finding the tutoring helpful and 89% agreeing that Oscar helped them to revise. 91% of the sample said that they would use the Oscar CITS resource, with 86% stating they would use Oscar to support classroom tutoring and 77% saying they would use Oscar instead of learning from a book. A surprising 50% of the sample said they would use Oscar in place of attending face-to-face tutorials. There was a 21% mean test score improvement after tutoring for students who did not achieve full marks in the initial test.

6 Conclusions and Further Work

This paper has presented a novel conversational intelligent tutoring system called Oscar, which implicitly predicts and adapts to a student's learning style during a tutoring conversation. Oscar employs a conversational agent to intelligently lead an online tutorial, mimicking a human tutor in offering students learning material adapted to their learning styles, individualised problem solving support and intelligent solution analysis. A CITS which personalises tutoring by dynamically predicting and adapting to learning styles could improve the effectiveness of a student's learning experience and help to boost confidence. Effective, personalised online tutoring could support class-based courses and widen access through distance learning.

The results of the initial study showed that students whose learning path adapted to their learning styles achieved on average 12% more correct answers than those students presented with learning material not matched to their learning styles. With regards to Oscar's conversational tutoring, the results have shown that the subjects did value the online Oscar CITS and that Oscar's tutoring seemed to help learning and improved test scores by 21% on average. It can therefore be concluded that using Oscar has helped provide students with a positive learning experience.

In future, it is planned to incorporate the tutor material categories into a toolkit to speed up the development of an adaptive CITS.

Acknowledgement. The authors thank Convagent Ltd for the use of the InfoChat conversational agent and PatternScript scripting language.

References

- [1] Brusilovsky, P., Peylo, C.: Adaptive and Intelligent Web-based Educational Systems. *Int. J. Artificial Intelligence in Education* 13, 156–169 (2003)
- [2] D'Mello, S., Lehman, B., Sullins, J., Daigle, R., Combs, R., Vogt, K., Perkins, L., Graesser, A.: A Time for Emoting: When Affect-Sensitivity Is and Isn't Effective at Promoting Deep Learning. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 245–254. Springer, Heidelberg (2010)
- [3] Graesser, A., Chipman, P., Haynes, B.C., Olney, A.: AutoTutor: An Intelligent Tutoring System With Mixed-Initiative Dialogue. *IEEE Trans. Education* 48(4), 612–618 (2005)

- [4] Popescu, E.: Adaptation provisioning with respect to learning styles in a Web-based educational system: an experimental study. *Journal of Computer Assisted Learning* 26, 243–257 (2010)
- [5] Felder, R., Silverman, L.K.: Learning and Teaching Styles in Engineering Education. *J. Engineering Education* 78(7), 674–681 (1988)
- [6] Cha, H.J., Kim, Y.S., Park, S.-H., Yoon, T.-b., Jung, Y.M., Lee, J.-H.: Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 513–524. Springer, Heidelberg (2006)
- [7] Klasnja-Milicevic, A., Vesin, B., Ivanovic, M., Budimac, Z.: E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education* 56, 885–899 (2011)
- [8] Chi, M.T.H., Siler, S., Jeong, H., Yamauchi, T., Hausmann, R.G.: Learning from human tutoring. *Cognitive Science* 25, 471–533 (2001)
- [9] Woo Woo, C., Evens, M.W., Freedman, R., Glass, M., Seop Shim, L., Zhang, Y., Zhou, Y., Michael, J.: An intelligent tutoring system that generates a natural language dialogue using dynamic multi-level planning. *Artificial Intelligence in Medicine* 38, 25–46 (2006)
- [10] Mitrovic, A.: An Intelligent SQL Tutor on the Web. *Int. J. Artificial Intelligence in Education* 13, 171–195 (2003)
- [11] Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., Ullrich, C.: ActiveMath: A web-based learning environment. *Int. J. Artificial Intelligence in Education* 12(4), 385–407 (2001)
- [12] Ammar, M.B., Neji, M., Alimi, A.M., Gouarderes, G.: The Affective Tutoring System. *Expert Systems with Applications* 37, 3013–3023 (2010)
- [13] Leontidis, M., Halatsis, C.: Integrating Learning Styles and Personality Traits into an Affective Model to Support Learner’s Learning. In: Spaniol, M., Li, Q., Klamma, R., Lau, R.W.H. (eds.) *ICWL 2009. LNCS*, vol. 5686, pp. 225–234. Springer, Heidelberg (2009)
- [14] Latham, A.M., Crockett, K.A., McLean, D.A., Edmonds, B.: Predicting Learning Styles in a Conversational Intelligent Tutoring System. In: Luo, X., Spaniol, M., Wang, L., Li, Q., Nejdil, W., Zhang, W. (eds.) *ICWL 2010. LNCS*, vol. 6483, pp. 131–140. Springer, Heidelberg (2010)
- [15] Latham, A., Crockett, K., Bandar, Z.: A Conversational Expert System Supporting Bullying and Harassment Policies. In: *ICAART 2010*, pp. 163–168 (2011)
- [16] Pudner, K., Crockett, K.A., Bandar, Z.: An Intelligent Conversational Agent Approach to Extracting Queries from Natural Language. In: *Proc. WCE Int. Conf. Data Mining and Knowledge Engineering*, pp. 305–310 (2007)
- [17] Michie, D.: Return of the Imitation Game. *Electronic Transactions on Artificial Intelligence* 6, 203–221 (2001)
- [18] Li, Y., Bandar, Z., McLean, D., O’Shea, J.: A Method for Measuring Sentence Similarity and its Application to Conversational Agents. In: *Proc. FLAIRS 2004*, pp. 820–825 (2004)
- [19] Khoury, R., Karray, F., Kamel, M.S.: Keyword extraction rules based on a part-of-speech hierarchy. *Int. J. Advanced Media and Communication* 2(2), 138–153 (2008)
- [20] Latham, A.M., Crockett, K.A., McLean, D.A., Edmonds, B., O’Shea, K.: Oscar: An Intelligent Conversational Agent Tutor to Estimate Learning Styles. In: *Proc. IEEE World Congress On Computational Intelligence 2010*, pp. 2533–2540 (2010)

An Information Recommendation Agent on Microblogging Service

Taketoshi Ushiana and Tomoya Eguchi

Faculty of Design, Kyushu University,
4-9-1 Shiobaru, Minami-ku, Fukuoka 815-8540, Japan
{ushiana@, eguchi@gospel.aid.}design.kyushu-u.ac.jp
<http://www.design.kyushu-u.ac.jp/~ushiana/>

Abstract. In this paper, we introduce an agent for recommending information to a user on Twitter, which is one of the most popular microblogging services. For recommending sufficient information for a user, it is important to extract automatically user's interest with accuracy and to collect new information which interests and attracts the user. The agent that is introduced in this paper extracts automatically user's interests from tweets on the timeline of the user and finds the web sites that would provide new information which interests the user from the tweets. The agent selects recommending information from the web sites and posts it on the user's timeline. Experimental results show that our agent is able to recommend sufficient information for users of Twitter in a natural manner.

Keywords: Recommendation, microblogging service, Twitter, content-based filtering, automatic profile composition.

1 Introduction

Today, large amount of information can be found on the Web. Its volume is so large that it is difficult for people to find information which interests and/or attracts them. Based on such background, the importance of the systems which support people to obtain useful and interesting information for them is increasing. In a recent few years, microblogging services such as Twitter become very popular [1]. Users of microblogging services post short messages which are limited up to about one or two hundreds characters. In this paper, we introduced a method for recommending information by means of Twitter.

Twitter has some characteristic features which traditional blogging services do not have. Firstly, the length of a text message posted by users on Twitter is limited to 140 characters so users can post their messages lightheartedly. Messages which are posted by a user may contain various and amount of hints which are useful for conjecturing some interests of the user. Secondly, Twitter provides a function for gathering the messages that are posted by people who are expected to provide information which are useful for a user. Such messages are listed in a timeline format, where messages flow according to their posted date time. Users

do not expect that all messages on their timelines are valuable for them. We think that the timeline format is suitable for recommendation. This is because relatively useless information which is recommended on their timelines would be acceptable for them.

In this paper, we propose a method to recommend information which a user on Twitter would be interested in based on his/her tweets posted previously. Using this method, users can obtain personalized attractive information based on daily tweeting without giving any explicit queries.

This paper consists of the follows sections. Section 2 gives an approach of our method. Section 3 describes details of the method. Section 4 shows some experimental results and discusses the effectiveness of the method. Section 5 describes conclusion and future directions.

2 Approach

Generally recommendation methods can be categorized into two types: content-based filtering approach and corroborative filtering approach[2].

In content-based filtering approach[3], a profile of a user is generated based on analyzing the items that the user has selected. The profile represents interests of the user. A recommendation system searches information or items which matches to the profile and recommends them. In order to compose user profiles, various techniques have been proposed. One of popular approaches is based on the vector-space model[4]. This approach uses a vector for representing a user profile. Furthermore it uses a vector for representing features of an information item which is a candidate for recommendation. Recommendation systems based on the vector-feature model filter information items to be recommended based on the similarity on a pair of a user profile and an information item. If an information item is similar to the user profile, it is recommended to the user.

In collaborative filtering approach[5,6,7], recommendation systems do not have to use features of information items which are candidates for recommendation. The basic idea of collaborative filtering is to recommend to a user A the information items that are evaluated to high scores by a user B if the profile of the user A is similar to the profile of the user B. Practically, recommendation systems calculate the similarity of every pair of users based on evaluations of each user for information items and recommend to a user one or more information items which are well evaluated by the other users who can be considered to be similar to him/her.

Content-based filtering techniques can hardly be adaptable to items, such as movies or shopping products, from which it is difficult for computers to extract their highly semantic features automatically. This is because that it is necessary to compose a sufficient feature vector which represents highly semantic characteristic features of each recommendation item for content-based filtering. Collaborative filtering techniques can be adaptable to such items because collaborative filtering techniques do not use features of recommendation items directly but use evaluations on the recommendation items by users. On the other hand,

collaborative filtering techniques have some drawbacks. One of them is that collaborative filtering techniques cannot recommend new items which have not been evaluated by any user. Furthermore, the precision of recommendation might be low if the numbers of evaluations for recommendation items by users are small.

The objective of the method proposed in this paper is to recommend information items which might interest or attract a user from the information items which are newly uploaded on the Web. We cannot suppose that such information items have been evaluated by many enough users for effective recommendation by means of corroborative filtering techniques. Therefore we employ a content-based filtering technique for our objective.

When using content-based filtering techniques, it is important to compose a sufficient profile of a user. Some techniques have been proposed to compose sufficient profiles. One major approach is to compose user profiles based on the keywords that are given explicitly by users and another major approach is based on users' behaviors such as their access histories of web pages. The keyword based approach requests users to do a kind of extra load and we cannot always obtain sufficient profiles about them. Moreover, even if a user composes his/her sufficient profile once, the user has to update manually his/her profile when new interesting topics appear on the Web and/or the user becomes interested in different topics. It is additional extra load for the user. When users have to do extra load for updating their profiles, their profiles would leave insufficient and it causes less precision of recommendation.

The objective of our system is to recommend a user, who utilizes microblogging services routinely, information items which would interest and/or attract the user without explicit specification of his/her profile. The system extracts interests of the user automatically by means of analyzing the tweets that appear in the timeline of the user. Furthermore, the system extract references to the web sites from the tweets. We considered that the extracted references would be candidates for good information sources where valuable information items for the user will be uploaded. The system observes uploads of the extracted references and the updates become candidates of information items for recommendation. The system checks whether each information item is valuable for the user based on the extracted user's interests and decides information items to be recommended. Finally the system posts the recommending information items on the timeline of the user. This method enables users to obtain information items which would interest and/or attract them in a natural manner without any explicit operation for specifying their interests and/or preferences.

3 Information Recommendation Agent

This section introduces an agent for recommending information items on a microblogging service. This agent is mainly two components: the module for extracting interests of a user and the module for gathering and selecting information items to recommend for the user based on his/her interests. Fig. 1 shows the data flow of our recommendation system. Tweets which are posted

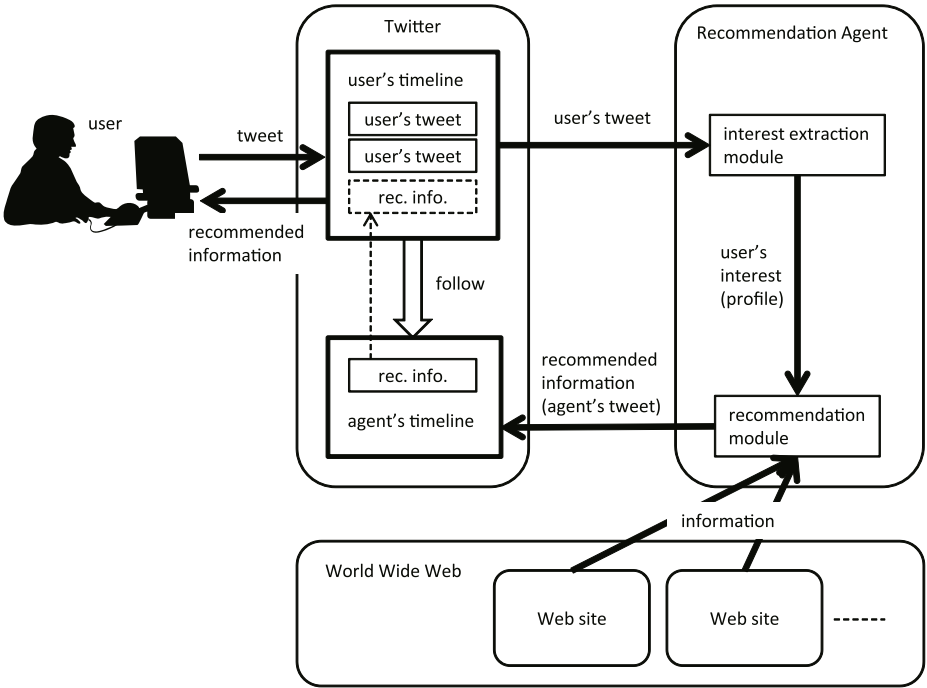


Fig. 1. Data flow of the recommendation system

by a user are fetched by his/her recommendation agent. The interest extraction module analyzes the tweets and extracts his/her interests. Then the recommendation module gathers information items from some web sites and select information items which are recommended for the user based on his/her extracted interests. The recommending information items are formatted as tweets and posts them on the agent's timeline, which has been followed by the user automatically. The posted agent's tweets, which are the recommendation information, are displayed on user's timeline, so the user can obtain the recommended information.

3.1 Extracting User's Interests

Tweets and User's Interests. Our method tries to extract interests of a user by means of the tweets that the user has posted previously. This approach is employed based on the assumption that tweets of a user reflects his/her interests. On Twitter, a user would report events which happen around the user or personal simple opinions which the user has as tweets. The person who reads tweets which are posted by a user is not specified by the user. Most of tweets are able to be read by anyone who wants to. Twitter provides a function to receive the tweets posted by a target person. This function is named as *follow*. On Twitter, users would

try to gather information items which interest and attract them by following other users who are expected to provide such information items. Because of this system, many tweets are not expected to be replayed by other person. A user can stop following another user when he/she wants to. Because Twitter has those features, the initiative of communication on Twitter is controlled by readers. Therefore, users on Twitter are easy to post their natural opinions as tweets without considering reactions for them by their readers.

According to the above features of Twitter, we considered that tweets of a user reflect interests of the user. Therefore, in our method, the system extracts interests of a user from tweets by means of analyzing them and recommends information items based on the extracted interests.

Extracting User's Interests from Tweets. Our method for extracting interests of a user from tweets consists of following steps:

1. obtaining tweets of users,
2. morphological analysis on the tweets,
3. selecting keywords, and
4. expanding keywords.

Firstly the agent collects all tweets from the timeline of a user whom information is recommended to. Two types of tweets exist in the timeline: the tweets posted by the user and the tweets posted by others whom the user follows. Then the agent decomposes the collected tweets into words by morphological analysis and extracts nouns. They are candidates for the keywords that represent interests of the user.

Each extracted nouns is assigned its weight. In searching documents, the TF-IDF method has been used conventionally for calculating the weight of a term in a document in a document set [8]. The TF-IDF value of a term t in a document d is defined as the following formula:

$$\text{tfidf}(t, d) = \frac{ft_{d,t}}{\sum_i ft_{d,i}} \cdot \log \frac{|D|}{fd_t}, \quad (1)$$

where $ft_{d,t}$ is the number of occurrences of the term t in the document d , D is the target set of documents, and fd_t is the number of documents which contain the term t in D . The TF-IDF method is designed based on the assumption that a term which occurs frequently in a document would represent an important concept in the document and a term which is rarely contained in documents would represent a characteristic concept in the document set.

For our objective, it is required to extract interests of a user from the tweets on the timeline of the user. The TF-IDF method cannot be applied directly for our purpose because it represents importance of a term on a document in a document set and many tweets (documents) would be exists on the timeline of the user. Therefore, we introduce a novel weighing schema called TF-IUF. The TF-IUF value of a term t for a user u is defined as following formula:

$$\text{tfiuf}(t, u) = \frac{\sum_{tw \in TW_u} ft_{tw,t}}{\sum_i \sum_{tw \in TW_u} ft_{tw,i}} \cdot \log \frac{|U|}{fu_t}, \quad (2)$$

where TW_u is a set of tweets on the timeline of the user u , and U is a set of users, and fu_t is the number of users whose tweets contain the term t .

We employ the TF-IUF value for weighing the importance of a term for a user and consider that it reflects the degree of interest of the user about the topic represented by the term. The TF-IUF method is designed based on the assumption that a frequently used term in the tweets on the timeline of the user would concern a topic he/she is interested in and a term which are seldom appeared on the timelines of others represent a characteristic interest about the user.

Keywords which represent interests of a user are selected based on the weights of terms. Top n terms in the list that is ordered by their weights are treated as the keywords. n is a threshold given by a user.

Some of the selected keywords might be too specialized for representing an aspect of user's interests. Therefore, each keyword is expanded in order to recommend information in areas which the user is interested in. For the keyword expansion, we use Yahoo web search API¹. This API has function to return the category of a term.

3.2 Recommending Information

The agent recommends information for a user based on the extracted interests of the user. The information recommendation consists of the following steps:

1. finding information sources (web sites),
2. obtaining information items from the information sources,
3. selecting information items to be recommended,
4. composing recommendation messages, and
5. posting the messages as tweets for the user.

In order to recommend information to a user, the agent has to find information sources which have possibility to provide information items attracting the user. In order to find such information sources, our method utilize the URLs that are addressed in tweets on the timeline of the user. Many Twitter users sometimes include one or more URLs in their tweets for reference when they would like to introduce interesting topics found on the Web and to describe their opinions about them. The timeline of a user contains tweets of the other users followed by the user. Some of those tweets also contain URLs which provide information topics which have possibility to interest the user. Therefore, the agent extracts the URLs on the timeline of the user and utilizes them as clues to find information sources.

¹ <http://developer.yahoo.co.jp/webapi/search/websearch/v2/websearch.html>

The agent accesses the extracted URLs and checks whether each web site provides RSS feeds or not. RSS is a mechanism for notifying users of new information items on a web site. Users can obtain summaries and simple metadata of new information items from RSS feeds. When a web site provides RSS feeds, the agent periodically gets the RSS feed and obtains new information items, which are candidates for recommendation.

The agent examines whether each of the obtained information items should be recommended or not. Firstly, an information item is decomposed into nouns by morphological analysis. Then the feature vector of the information item, which is called as *information vector*, is composed based on a keywords vector of the user. When a keyword vector \mathbf{k} is given, the information vector of an information item d is defined as follows:

$$\mathbf{iv}_d = (\text{tfidf}(k_1, d), \dots, \text{tfidf}(k_n, d)), \quad (3)$$

where k_i is a i -th elements in the keyword vector \mathbf{k} .

An information item is decided whether it should be recommended by calculating the similarity between its information vector and the *interest vector* of the user. The interest vector represents interests of a user. The interest vector of a user u is defined as the following formula:

$$\mathbf{it}_u = (\text{tfiuf}(k_1, u), \dots, \text{tfiuf}(k_n, u)), \quad (4)$$

where k_i is a i -th elements in the keyword vector \mathbf{k} . The similarity of an information vector \mathbf{if}_d and an interest vector \mathbf{it}_u is defined as follows:

$$\text{sim}(\mathbf{if}_d, \mathbf{it}_u) = \frac{\mathbf{if}_d \bullet \mathbf{it}_u}{|\mathbf{if}_d| \cdot |\mathbf{it}_u|}. \quad (5)$$

If the similarity is greater than the threshold t , the information item is decided to be recommended.

For every information item, the similarity between it and the interest vector of the user is calculated and decided whether it should be recommended or not. Then a tweet is generated for notifying each recommendation information item. The tweet consists of title and its URL, which can be obtained from RSS. When the URL is too long, it is shortened by a URL shortening service. The generated tweet is posted on Twitter by the account of the agent and it is listed on the timeline of the user. If the user is interested in the tweet, he/she can click the URL on it and the web page of the recommended information item is displayed.

4 Evaluation

In order to evaluate the effectiveness of the proposed method, we implemented a prototype of the agent and conducted some user studies.

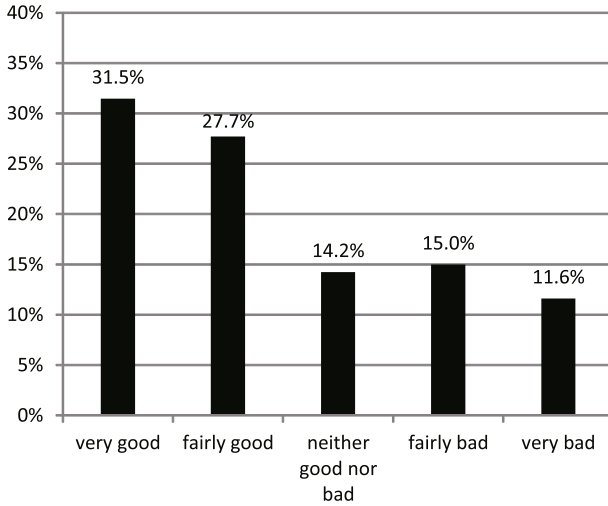


Fig. 2. Results of evaluation for extracted keywords

4.1 Prototype System

A prototype system of the information recommendation agent was implemented on a personal computer, which runs Windows 7 OS and has Intel Core 2 Duo E4500 CPU and 4GB memory. The program of the agent is coded by means of the Perl programming language with Twitter REST API and Twitter Streaming API².

The agent has its account on Twitter. Thus, a user can receive recommended information from the agent by just following the account. When the agent is followed by the user, the agent extracts interests of the user from his/her timeline, then recommends information items as tweets for the user. The agent updates the interests of the user when a new tweet is posted. The agent recommends information once in a day.

4.2 Experimental Results

In order to evaluate the effectiveness of the proposed method, we conducted two types of user studies. One is to evaluate whether the proposed interest extraction method is able to extract keywords that reflect the interests of a user. The other is to evaluate whether the proposed information recommendation method is able to find and select information items that interest and/or attract a user.

Extraction of User's Interests. We made the agent extract interests of an experimental subject from his/her timeline and selected top-15 keywords based on the TF-IUF schema, then asked the subject to evaluate each of the keywords by means of 1 to 5 scale from a view of reflecting his/her interests. The number

² <http://dev.twitter.com/doc>

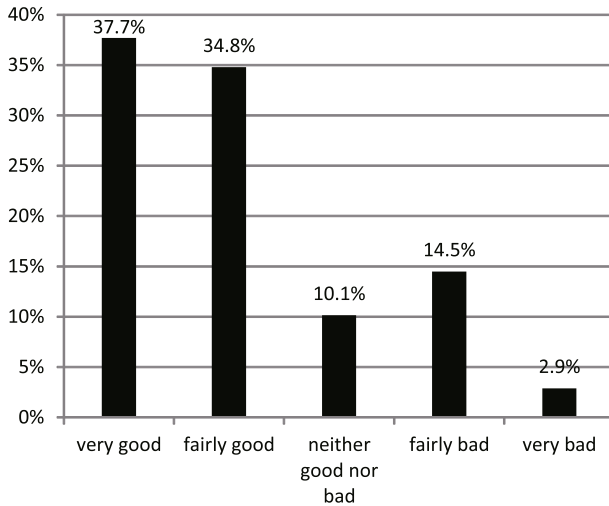


Fig. 3. Results of evaluation for recommended information items

of the subjects is seven. Fig. 2 shows the results of this study. The average score of the evaluations is 3.5.

Recommendation of Information. We made the agent find and select top-10 recommending information items for an experimental subject based on the extracted URLs and interests from the timeline of the user, then asked the subject to evaluate each of the recommending information items by means of 1 to 5 scale from a view of it interests and/or attracts the user. The number of the subjects is seven. Fig. 3 shows the results of this study. The average score of the evaluations is 3.9.

4.3 Discussion

In the extracted keywords, 59.2% keywords were given positive grades (very good or fairly good) by the subjects. Many of the lowly graded keywords are names of friends of the subjects. By excluding such names the precision of extracting user's interests would be improved. Although such names are seldom contained the information items that are candidate for recommendation so we think that the effects of those noise for recommendation is not much.

In the recommended information items, 72.5% items are given positive grades (very good or fairly good). This ratio is greater than the ratio on the extracted keywords. This means that even if some insufficient keywords are selected, their effects on the performance of recommendation might be limited. For some of the recommended items that were given negative grades, the subjects could not understand the details of their contents from the tweets that introduce those items. This is because the technique for composing tweets for recommendation

is too simple, a tweet contains only the title and URL of the recommending information item. We think that more sufficient techniques for composing tweets for recommendation would improve the performance of recommendation.

5 Conclusion

In this paper, we introduced an information recommendation agent for a user on Twitter. This agent observes tweets of a user and his/her companions (followers) and extracts his/her interests and web sites that have possibility to provide information items attracting the user. We designed the TF-IUF method for extracting user's interests. This method is able to weight a term from a view of user's interest. The agent accesses the web sites periodically and selects recommending information items for the user. Then the agent posts tweets about the recommending information items on the timeline of the user. Our prototype of the agent starts recommending information by only following its account of Twitter. Results of the user studies on the prototype show that our method is effective for recommendation.

We have a plan to improve the precision of recommendation based on feedback of users such as clicks on recommended URLs. Moreover, we would like to extract interests of communities based on following networks on Twitter.

References

1. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Twitter power: Tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.* 60, 2169–2188 (2009)
2. Hanani, U., Shapira, B., Shoval, P.: Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction* 11, 203–259 (2001)
3. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 1349–1380 (2000)
4. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* 18, 613–620 (1975)
5. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. *Adv. in Artif. Intell.* 2009, 4:2–4:2 (2009)
6. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 175–186. ACM, Chapel Hill (1994)
7. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995*, pp. 210–217. ACM Press/Addison-Wesley Publishing Co, New York, NY, USA (1995)
8. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1998*, pp. 275–281. ACM, New York (1998)

Movement Algorithms for Refugee Agents for Virtual Disaster Simulation Systems

Koichi Asakura¹, Hiroki Aoyama¹, and Toyohide Watanabe²

¹ School of Informatics, Daido University,
10-3 Takiharu-cho, Minami-ku, Nagoya, 457-8530 Japan
asakura@daido-it.ac.jp, syu.hironegi@gmail.com

² Department of Systems and Social Informatics, Graduate School of Information
Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
watanabe@is.nagoya-u.ac.jp

Abstract. In this paper, we propose movement algorithms for refugee agents in disaster areas for developing simulation systems of virtual disaster areas. Usually, refugees in disaster areas move to shelters for evacuation. However, such evacuation activities vary depending on the types of refugees. In our algorithms, we provide two features for representing the types of refugees: their possession of map information and their state of mind. Experimental results show that our proposed movement algorithms for refugees can express the activities of refugees correctly and thus are suitable for virtual disaster simulation systems.

Keywords: movement algorithm of agent, simulation, disaster area.

1 Introduction

In disaster situations such as a big earthquake, a robust communication system is needed to gather and share information on disaster areas [1,2]. In such situations, however, mobile phones and wireless LAN networks cannot be used since communication infrastructures such as base stations and WiFi access points may be broken or malfunction in disaster areas. Thus, we have to develop wireless network technologies that do not depend on such communication infrastructures.

For communication systems in disaster areas, mobile ad-hoc network (MANET) technologies have attracted great attention recently [3,4,5]. The MANET is defined as an autonomous network by a collection of nodes that communicate with each other without any communication infrastructures. Nodes in the MANET communicate with each other and generate communication paths cooperatively. In order to deal with topology changes in networks caused by the mobility of nodes, nodes restructure network conductivities with neighboring nodes autonomously and locally. Therefore, since the MANET does not require any communication infrastructures, it is suitable for communication systems in disaster areas.

Our research goal is to develop an effective MANET-based information sharing system suitable for disaster areas. We have thus already developed the ad-hoc unicursal protocol (AUP) for such systems [6,7]. In this protocol, unicursal paths

are generated in order to decrease the frequency of communication, which leads to greater longevity of networks and terminals because of lower battery consumption. Also, we have developed a system for simulating disaster areas, especially for a big earthquake [8]. In this simulation system, damaged roads that prevent people from passing through faster and collapsed buildings that also curtail people's mobility are generated on the basis of hazard maps [9,10,11] and related algorithms [12].

In this paper, we propose movement algorithms for refugee agents for virtual disaster simulation systems. Disaster areas contain many types of refugees. We model the activities of refugee agents by using two features: their possession of map information of disaster areas and their state of mind. By using our algorithms, the simulation systems can provide realistic movement of refugees, which enables us to evaluate communication systems in disaster areas effectively.

The rest of this paper is organized as follows. Section 2 describes our approach for modeling movement of refugee agents and related work. Section 3 gives movement algorithms for refugees. Section 4 explains our experiments. Finally, Section 5 concludes this paper and gives our future work.

2 Approach

In this section, we describe our approach for developing algorithms for refugee mobility. Section 2.1 summarizes the kinds of refugees and considerations for the algorithms. Section 2.2 expresses related work.

2.1 Movement of Refugees

Disasters such as big earthquakes occur suddenly. People suffer these disasters wherever they happen to be at that time. Needless to say, people can be struck by a disaster in a place they do not know. Therefore, disaster areas contain many types of refugees. For example, some people can move to the nearest shelter calmly, while some people do not know the locations of shelters clearly and move around randomly. Also, some panic while others stay calm. In disaster areas, important features that influence the behavior of refugees are as follows:

- refugees' map information for disaster areas and
- refugees' state of mind.

Map Information. The behavior of refugees highly depends on their map information of disaster areas. This is because map information such as location of shelters, appropriate roads to take to the shelters, and so on is essential for successful evacuations from disasters. If refugees encounter disasters in the place they ordinarily live, they know the location of shelters and thus can take the appropriate roads to them. However, if they meet with disasters in unfamiliar places such as on holiday, and so on, they usually have little knowledge about locational information of disaster areas. Thus, refugees may be confused and

fail to take appropriate actions for evacuation. However, classifying refugees in accordance with map information is difficult. Some people may know the location of shelters but not the ways to them.

In our simulation system, we provide the following three types of refugees in accordance with their quantities and qualities of map information of a disaster area.

Locals: they live in the place and thus are familiar with it. They know the location of the shelters and can choose appropriate roads to them. They do not lose their way during evacuation.

Newcomers: they live in the place, but do not know precise locations of the shelters. They cannot select appropriate roads for evacuation.

Outsiders: they do not live in the disaster areas and do not know the location of shelters. Thus, they always lose their way and have great difficulty in getting to the shelters.

State of Mind. When people face abnormal situations such as disasters, their confusion makes it impossible for them to take actions they can easily perform in everyday situations. Thus, in order to model the mobility of refugees, we have to pay attention to their state of mind as well as their map information. In our simulation system, we model the following two states of mind:

- a normal state and
- a panic state.

When refugees are in a panic state, they cannot move correctly. For example, they cannot choose appropriate roads, they do not know where they are, and they may move around in the same place. In our simulation system, in order to realize the above difference, the amounts of memory refugees have are varied. When refugees are in a panic state, their amount of memory decreases and thus they cannot remember roads that have previously used.

2.2 Related Work

There are many research projects on simulating communication systems. The GloMoSim project [13] and the JiST/SWANS project [14,15] are research projects for developing and evaluating the MANET protocols. However, these systems only provide very simple mobility algorithms such as a random way point mobility model and therefore cannot be used for evaluation from disaster areas. For a vehicular mobility model, the street random way point (STRAW) mobility model has been proposed [16]. In this model, nodes move to the destination point along the roads. However, this model is provided for a vehicular ad-hoc network (VANET) and thus cannot be adapted to mobility of refugees as it is. Also, there are many agent-based simulation projects for refugee evacuation [17,18]. However, these projects mainly focus on microscopic simulation, i.e. a movement algorithm in a corridor to avoid collision with other refugees.

3 Algorithms

As shown in Section 2, we provide three types of people with respect to map information and two types of people with respect to state of mind for the mobility of refugees. Table 1 shows all types of refugees. It is assumed that locals do not panic since they have in-depth information of the disaster areas. Thus, we have to develop five types of mobility algorithms for refugees. Here, we adopt the Dijkstra’s algorithm [9], which calculates the shortest path to shelters, as the mobility algorithm for locals. From now on, we describe algorithms for the other four types of mobilities.

3.1 Refugee Agents

Refugee agents have the following properties for the algorithms.

Type: this states the type of a refugee: a local, a newcomer or an outsider.

State of mind: this expresses whether a refugee is in a normal or panic state.

Memory of roads: this stores the history of roads to take. The amount of memory, i.e. how many roads to take can be remembered, depends on the refugee’s state of mind.

Shelter location: this denotes the destination shelter. This property is used for newcomers.

Direction: this describes the direction in which a refugee moves. The values of this property are North, South, East, and West. This property is used for outsiders.

“Type” is defined first and does not change during evacuation. “State of mind” is also static in our system. Dynamic change in refugees’ state of mind is future work.

3.2 Algorithms for Newcomers

In our simulation system, it is assumed that newcomers know the location of shelters roughly but not the appropriate roads to take. Thus, newcomers know the rough direction of shelters and choose the roads in accordance with the angle between the road and the direction of shelters.

While moving to the shelter s , a refugee chooses the next road by the following algorithm when the refugee arrives at an intersection a .

Table 1. The type of refugees

	In a normal state	In a panic state
Locals	○	—
Newcomers	○	○
Outsiders	○	○

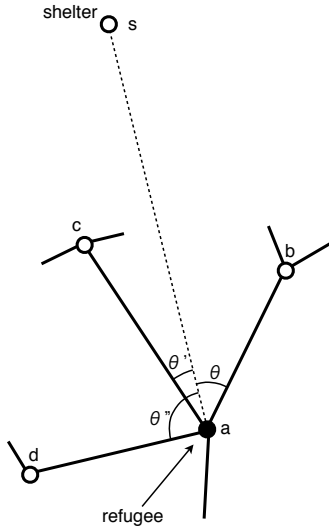


Fig. 1. Road selection method for newcomers

1. Choose the roads connected to the intersection a . Chosen roads are denoted as vectors: r_1, \dots, r_n .
2. Calculate the vector representing the direction from the intersection a to the shelter s : r_s .
3. Chosen roads r_1, \dots, r_n are sorted in accordance with the angle between r_i and r_s .
4. The top-rank road is chosen as the next road to take.

Figure 1 shows the road selection method in this algorithm. For each road \vec{ab} , \vec{ac} , and \vec{ad} , the angle between the road and \vec{as} is calculated. In this situation, the road \vec{ac} is selected. In Step 1, if a destination of a road is included in a refugee's memory, i.e. the refugee has already been to the destination before, this road is omitted.

3.3 Algorithms for Outsiders

Outsiders know neither the location of shelters nor the appropriate roads used for evacuation. Thus, at first, outsiders move randomly. When they reach an intersection, they select the direction in which they next move. In order to select the direction, we use the number of refugees on the roads. Namely, the number of refugees on the roads is counted for all roads connected to the intersection. Then, an outsider selects the direction to the road on which the most refugees are moving. This process expresses that outsiders follow other refugees since they do not know where to evacuate.

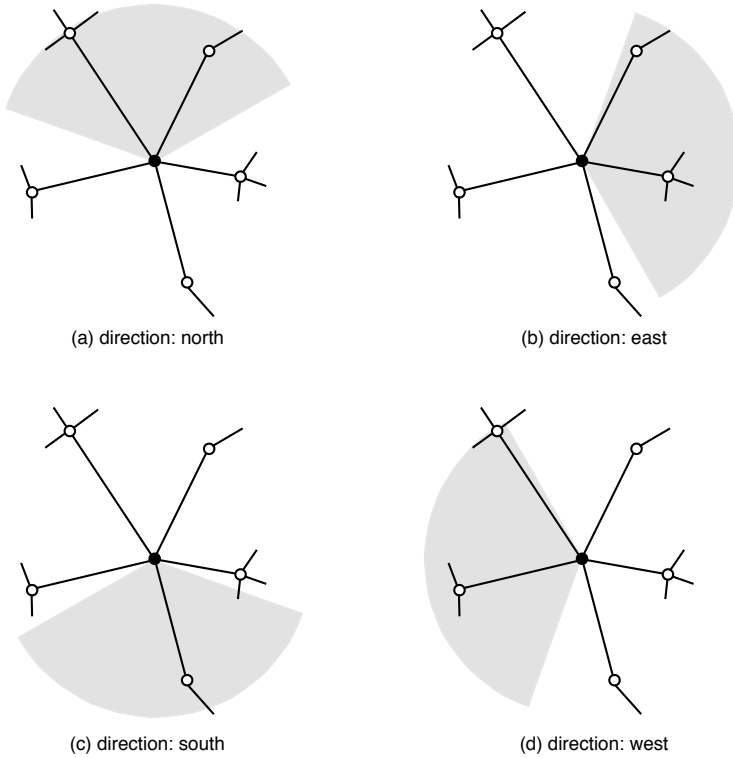


Fig. 2. Road selection method for outsiders

An outsider chooses the next road by the following algorithm when the refugee arrives at an intersection c .

1. Choose the roads connected to the intersection c . Chosen roads are denoted as vectors: r_1, \dots, r_n .
2. Candidate roads are selected from the above roads in accordance with the direction in which the refugee is going. We use fan-shaped regions for selection of roads. Figure 2 shows the fan-shaped regions for each direction.
3. The next road to take is chosen from the candidate roads.
4. When refugees reach a dead-end, they move randomly for a certain time and then select the direction again.

Like Step 1 in the newcomer's algorithm, if a destination of a road is included in a refugee's memory, i.e. the refugee has already been to the destination before, this road is also omitted in Step 1.

4 Experiments

We conducted an experiment for evaluating movement algorithms of refugee agents. This section describes the experimental results.



Fig. 3. Simulation area

4.1 Simulation Settings

In the experiments, we provided a virtual disaster area 2.0 kilometers wide and 1.5 kilometers high. Figure 3 shows the simulation area, in which shelters are denoted as circles.

In the experiments, we evaluated five types of movement algorithms: algorithms for locals, for newcomers in a normal state, for newcomers in a panic state, for outsiders in a normal state, and for outsiders in a panic state. The number of refugees of each type was 100. Thus, 500 refugees were deployed on the road randomly and tried to evacuate. We performed ten experiments. In order to evaluate algorithms, we compare the following numbers.

- The number of intersections passed. This indicator shows the time it takes to arrive at the shelters.
- The number of times the same intersection is passed.
- The number of times a dead-end is selected.
- The number of times the optimal road is selected. Here, “optimal” means that the road is included in the shortest path to the shelters calculated by Dijkstra’s algorithm [19].

4.2 Experimental Results

Table 2 shows the experimental results. This table shows the average numbers per refugees. This ratio of optimal roads selected is also described. From these results, we can clearly observe three types of refugees with respect to refugees’

Table 2. Experimental results

Refugee type	The number of intersections passed	The number of times the same intersection is passed	The number of times a dead-end is selected	The number of times the optimal road is selected	The ratio of optimal roads selected [%]
locals	16.55	0	0	16.55	100
newcomers in a normal state	22.77	0.25	0.08	16.10	70.71
newcomers in a panic state	27.88	0.40	0.10	17.70	63.49
outsiders in a normal state	94.76	17.96	2.97	42.05	44.37
outsiders in a panic state	117.07	31.74	4.77	51.58	44.06

map information. Locals can move to the shelter with the shortest path since they select optimal roads 100% of the time. Also, newcomers can reach the shelters faster than outsiders. Outsiders move in the same direction for a while once they determine the direction of evacuation. Thus, outsiders reach dead-ends more times than newcomers. In terms of state of mind, refugees in a panic state takes a lot of time to reach the shelters. Refugees in a panic state pass the same intersection much more than refugees in a normal state. Especially, outsiders in a panic state pass the same intersection about twice as much as outsiders in a normal state. These results show that confusion of refugees can be denoted as the quantity of memory. From the above consideration, we can clarify that our proposed movement algorithms for refugee agents are suitable for simulation systems of virtual disaster areas.

5 Conclusion

In this paper, we propose movement algorithms of refugee agents for developing a simulation systems for virtual disaster areas. In the algorithms, we dealt with two properties in order to express the characteristics of refugees: map information and state of mind of refugees. Map information denotes the degree the refugees understand the situation, and state of mind denotes the degree the refugees are confused. We provided five types of refugees and evaluated the evacuation activities of each refugee by using the simulation system for virtual disaster areas. Experimental results show that features of activities of each refugees can be realized by the two properties.

The movement algorithms described in this paper do not consider relationships between refugees. In real disaster areas, refugees often cooperate with each other in order to evacuate to shelters. Thus, for our future work, we have to develop multi-agent movement algorithms.

References

1. Midkiff, S.F., Bostian, C.W.: Rapidly-Deployable Broadband Wireless Networks for Disaster and Emergency Response. In: The 1st IEEE Workshop on Disaster Recovery Networks (DIREN 2002) (2002)

2. Meissner, A., Luckenbach, T., Risse, T., Kirste, T., Kirchner, H.: Design Challenges for an Integrated Disaster Management Communication and Information System. In: The 1st IEEE Workshop on Disaster Recovery Networks (DIREN 2002) (2002)
3. Toh, C.-K.: Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice-Hall, Englewood Cliffs (2001)
4. Murthy, C.S.R., Manoj, B.S.: Ad Hoc Wireless Networks: Architectures and Protocols. Prentice-Hall, Englewood Cliffs (2004)
5. Lang, D.: Routing Protocols for Mobile Ad Hoc Networks: Classification, Evaluation and Challenges. VDM Verlag (2008)
6. Asakura, K., Oishi, J., Watanabe, T.: An Ad-hoc Unicursal Protocol for Human Communication in Disaster Situations. In: The 2nd International Symposium on Intelligent Interactive Multimedia Systems and Services (KES-IIMSS 2009), pp. 511–521 (2009)
7. Asakura, K., Takikawa, H., Oishi, J., Watanabe, T.: An Ad-hoc Unicursal Protocol for Stable and Long-lived Communication Systems in Disaster Situations. International Journal of Knowledge and Web Intelligence 1(3/4), 154–168 (2010)
8. Asakura, K., Watanabe, T.: A Simulation System of Disaster Areas for Evaluating Communication Systems. In: The 2nd International Symposium on Intelligent Decision Technologies, pp. 495–506 (2010)
9. U.S. Geological Survey: Hazard Mapping Images and Data, <http://earthquake.usgs.gov/hazards/products>
10. The Geographical Survey Institute: Hazard Map Portal Site (in Japanese), <http://www1.gsi.go.jp/geowww/disapotal/index.html>
11. Nagoya City: Earthquake Maps for Your Town (in Japanese), <http://www.city.nagoya.jp/kurashi/category/20-2-5-6-0-0-0-0-0-0.html>
12. The Cabinet Office, government of Japan: A Manual for Assessment of Earthquake Damage (in Japanese), <http://www.bousai.go.jp/manual>
13. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. In: The 12th Workshop on Parallel and Distributed Simulations (PADS 1998), pp. 154–161 (1998)
14. Barr, R., Hass, Z.J., Renesse, R.: JiST: An Efficient Approach to Simulation Using Virtual Machines. Software – Practice and Experience 35(6), 539–576 (2005)
15. Barr, R., Hass, Z.J., Renesse, R.: Scalable Wireless Ad Hoc Network Simulation. In: Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks, pp. 297–311. CRC Press, New York (2005)
16. Choffnes, D.R., Bustamante, F.E.: Modeling Vehicular Traffic and Mobility for Vehicular Wireless Networks. Tech. Report NWU-CS-05-03, Department of Computer Science, Northwestern University (2005)
17. Uno, K., Kashiya, K.: Development of Simulation System for the Disaster Evacuation Based on Multi-Agent Model Using GIS. Tsinghua Science and Technology 13(1), 348–353 (2008)
18. Smith, J.L., Brokaw, J.T.: Agent-Based Simulation of Human Movements During Emergency Evacuations of Facilities. Structures Congress, 1–10 (2008)
19. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, Cambridge (2009)

Vehicle Routing Based on Traffic Cost at Intersection

Toyohide Watanabe and Chang Lijiao

Department of Systems and Social Informatics,
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
watanabe@is.nagoya-u.ac.jp

Abstract. Road/route searching problems have been investigated traditionally as one of the most basic/interesting issues for our human activities. In particular, the computer-supported route planning systems such as a car navigation system are usable in a wide range of applications. In this paper, we address a traffic route planning system to make the cost-effective driving time possible in comparison with the shortest route selection. Our idea is to compute the waiting time occurred by traffic signals at intersection and estimate the total driving time by the waiting time and ordinary running time for shortest route. In order to attain this objective, we distinguish the external link for road segment and the internal link which is three direction-dependent paths in inside of intersection: right-turn, straight and left-turn. Namely, the internal link is introduced to manipulate the waiting time in inside of intersection in addition to the conventional external link.

Keywords: vehicle navigation, traffic cost at intersection, Dijkstra's algorithm, computer simulation, road network, internal link, external link.

1 Introduction

Route searching problems have been investigated traditionally in various kinds of applications. The basic research point is to find out the cost-effective route such as the shortest path, the most fast driving route and so on. The famous searching method is Dijkstra's algorithm. Also, A*-algorithm is the best method to find out optimal solution. These methods which have been traditionally developed in a wide range of application domains until today focused on subjects about path searching with the minimum cost. Of course, various factors can be assigned as the cost estimation problems: one is the shortest length of total roads (or routes); and one is the shortest time to move fast; one is the cheapest fee to go to the destination; and so on. At least, to find out the cost-minimum route is not only the most basic problem in Mathematics but also the most interesting problems in social environments based on human activities.

In this paper, we address a problem to find out the cost-minimum driving route: the cost is estimated as the driving time. For example, some researches proposed car navigation systems by various types of methods which can recommend the optimal route from the guided vehicle [1-3]. These methods model a set of the corresponding regional roads as a road network with cost-attended links and then compute the

optimal route with the total minimum cost between origin point and destination point by Dijkstra’s algorithm or A*-algorithm. In [1], Ezawa et al. proposed the optimal method of traffic signals based on predictably crowded degree. In [2], Bazzan et al. computed the optimal route, using VICS information. Also, in [3] Yamashita et al. estimated the future-crowded situation on the basis of the current traffic information collected individually from running vehicles and proposed a cooperative route planning method. However, in these researches the waiting times generated by traffic signals at intersections over driving routes were not at all considered. In many cases, the vehicle movement is too largely dependent on traffic signals set almost at intersections, and it is important to navigate vehicles successfully from a viewpoint of cost-effectiveness. Our objective is to compute the waiting time generated by traffic signals at intersections on the basis of traffic situation and traffic signal switching time, and then estimate the total driving time. In order to attain this objective, our idea is to design the road network which can manage the waiting times of traffic signals at intersections as one of the cost factors so as to be able to compute the cost-effective route by Dijkstra’s algorithm.

2 Approach

2.1 Road Network

The road network, which has often been used in the existing route searching subject, is shown in Figure 1 and is composed of nodes, links and the connective relationships: the node corresponds to intersection, and the link represents road segment. Also, the connective relationships are the properties between road segments and intersections, and are the traffic features. Of course, these links are attached with their own particular costs. Our objective is to manipulate the waiting time, in which the running states of vehicles are disturbed by traffic signals at intersections, as the additional lost. This waiting time at intersections is varied, depending on the entrance lane and going direction. This is because the switching timing of traffic signals is different by the entrance lane and the waiting time is too strictly dependent on the entrance lane. In addition, when the number of vehicles is many in opposite lane even if the vehicle was on the same situation for the entrance lane, the waiting time is varied. This is because the waiting time on right-turn becomes too long. Thus, it is necessary to estimate the running time with respect to these two factors such as entrance lane and going direction when the vehicle passes by the intersection.

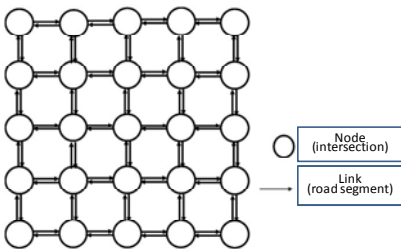


Fig. 1. Traditional road network

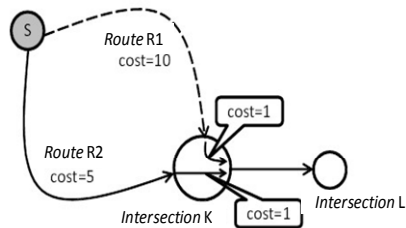


Fig. 2. Example of searching

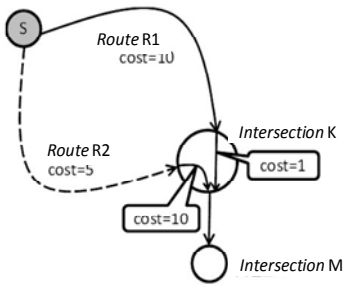


Fig. 3. Another example of searching

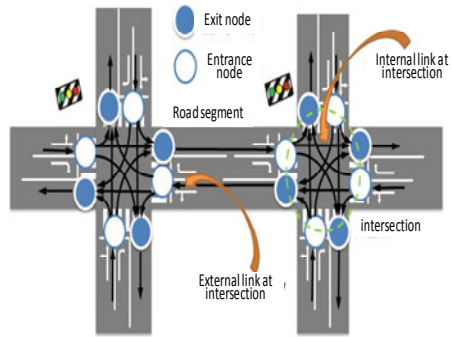


Fig. 4. Our road network model

In case that we want to assign the waiting time by traffic signal at intersection into the running time, the representative node must contain various types of costs at once since in the traditional model the intersection is not represented. Thus, Dijkstra’s algorithm is applied to search the shortest route totally after having composed individual shortest routes for each node [4]. Figure 2 and Figure 3 show the typical examples of shortest routes, when each node is respectively associated with such additional cost. In Fig.2, the route is selected through intersection *K* from start point *S* to the next intersection *L*. In this case, two different routes such as *R1* with cost 10 and *R2* with cost 5 are selected. Since the cost in intersection *K* is 1 the cost in the shortest route after having passed through intersection *K* is 6 along *R2*. However, in Fig.3 the cost in intersection *K* is different according to the going direction in order to go to another intersection *M*. The cost in intersection *K* from *R2* is 10, while the cost in intersection *K* from *R1* is 1. Thus, *R1* assures the shortest route. In order to compute the shortest route, each searched node must keep the minimum cost computed accumulatively from start node to current node. However, in the traditional road network the cost held temporarily in the current node cannot be computed if the next going direction were unclear. If our case in which each node has individually particular cost was applied to the estimation process, it is possible to again change the shortest route decided already for the currently searched node. Namely, the traditional model does not assume that the existing partial route is the shortest. Thus, in the traditional model it is not suitable to apply the dynamic searching method such as Dijkstra’s algorithm directly.

On the other hand, in our road network the model is looked upon as a pair of entrance lane gate and going direction, and also the costs corresponding to them are attached to the node. Figure 4 is our road network to be proposed in order to solve our objective. Our road network is definitely composed of the following elements:

- Exit node: node to be moved from inside of intersection to road segment;
- Entrance node: node to be accessible from road segments to inside of intersection;

- External link: link to associate cost of running time for corresponding road segment;
- Internal link: link to attend cost of running time to pass through intersection, and represent the going direction to be moved from inside of intersection.

Thus, in our case exit node and entrance node correspond respectively to exit and entrance of intersection. So, the road segments are represented as external links: the start point in the external link is an exit node; and the end point is an entrance node. The internal link corresponds individually to one of three going directions such as straight, left-turn and right-turn. The start in the internal link is an entrance node, and the end is an exit node. In our modeling road network, the costs are assigned to each link, but not so each node. Thus, Dijkstra’s algorithm is always applicable to the cost-estimation in our road network though the traditional networks cannot be applied directly.

2.2 Estimation of Link Cost

From a point of practical navigation view, individual costs attached with links must be specified time by time. So, the running time is too strongly dependent on the current traffic situation and the traffic signal switching timing. Figure 5 shows our navigation system ideally, which we assume. Of course, it is not impossible to set the networked infrastructure environment by current technologies though the realized sensors are not constructed completely over all city areas. Also, Figure 6 shows the processing flow between server installed in the traffic control center and clients to be manipulated by vehicles. The server collects traffic data or the related data from sensors preset on main important positions of road segments, computes the costs attended to external links and updates the traffic database constantly. On the other hand, each client searches the shortest route for the destination. In this case, the cost is computed, corresponding to the running timing on the shortest route when the shortest route tree is searched. Figure 7 shows such a searching process on the shortest route tree.

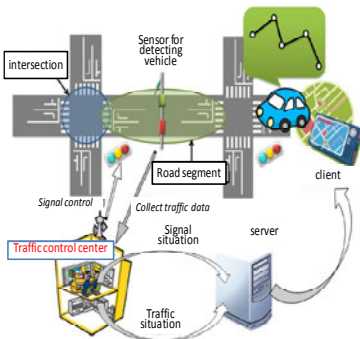


Fig. 5. Estimation of link cost

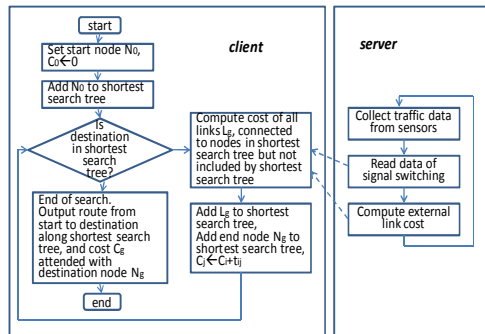


Fig. 6. Processing flow

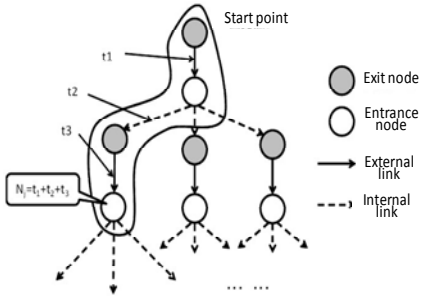


Fig. 7. Shortest route tree

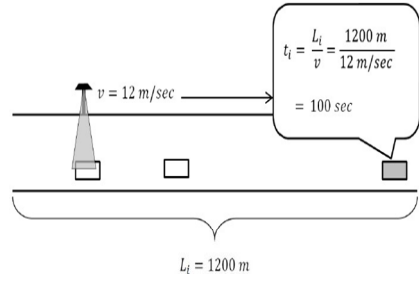


Fig. 8. Estimation of external link cost

3 Formulation

3.1 Preliminary

Our road network is represented as a graph G in Expression (1).

$$\begin{aligned}
 G &= (N_{en}, N_{ex}, L_{out}, L_{in}) & (1) \\
 N_{en} &= \{n_0^I, n_1^I, \dots\} \\
 N_{ex} &= \{n_0^O, n_1^O, \dots\} \\
 L_{out} &= \{l_{i,j}^O\} \quad (n_i \in N_{ex}, n_j \in N_{en}) \\
 L_{in} &= \{l_{j,i}^I\} \quad (n_j \in N_{en}, n_i \in N_{ex})
 \end{aligned}$$

The external link l^O is defined by two different attributes: pass-by speed v_l^O , and length L_l^O . The internal link l^I is composed of five attributes: pass-by speed v_l^I , going direction *course* I , cycle length of corresponding signal *cycle* I , split *split* I , and offset *offset* I . The entrance node n^I is attached with an attribute Q_n^I , which is the capacity of corresponding intersection. Also, the cost t_l^O in the external link l^O and the cost t_l^I in the internal link l^I are computed as we discuss in the next subsection. The estimation function in route searching is represented in Expression (2).

$$C(L_{route}^O, L_{route}^I) = \sum l^O \in L_{route}^O (t_l^O) + \sum l^I \in L_{route}^I (t_l^I) \quad (2)$$

L_{route}^O and L_{route}^I are respectively sets of external links and internal links which are selected from start node to destination node. Our objective is to compute Expression (2) practically and choose the minimum value for L_{route}^O and L_{route}^I .

3.2 Estimation of External Link Cost

As the volume of traffics interferers the moving times, the more the road segments are crowded, the longer the moving time is. We cannot estimate the moving time and arrival time even if we took into account the amount of route traffics between points. Additionally, we may loss a chance to select better route which reduces the running time, if we avoided selecting the crowded road segment. Until today, the method

which estimates the future running time on each segment using historical information is very standard and popular [5]. Using historical traffic information, we can model time-specific movement cost. Under this procedure, the moving time and arrival time to be predicted can be more reliable, and the practical driving time is furthermore reduced in comparison with the traditional approaches, which reduce the amount of traffics. In [6], Yu et al. proposed a method that observes the amount of traffics on real time and estimates the time-specific cost correctly. Figure 8 shows an example of cost computation on external link: the cost is composed by using vehicle’s speed, measured from vehicle sensor set at the side or near of road monitors and measurements, and road length. Running time cost of external link is shown in Expression (3).

$$t_i = L_i/v_i^{outside} \quad (l_i \in L^{outside}) \tag{3}$$

3.3 Estimation of Internal Link Cost

The internal link cost is defined as the movement cost from the entrance node to the corresponding exit node in the inside of intersection. When we estimate this internal link cost, we must pay attention to two factors. One is a waiting time by “red” signal under the traffic control and other is a delay time, derived from the interference of other vehicles. In case that the vehicle reaches to a row of other already waited vehicles before having arrived at the target intersection, the waiting time exhausted in the waiting queue is estimated as a part of internal link cost at intersection. Thus, we formulate this internal link cost as follows:

$$t_i^{inside} = t_k^{signal} + t_k^Q \quad (l_i \in L_{inside}) \tag{4}$$

Namely, our internal link cost is composed of waiting cost t_k^{signal} , in which the traffic signal is changed by the next “green” signal and passes with the cost t_k^Q at intersection. The method which computes t_k^{signal} is shown in Expression (5):

$$\begin{aligned}
 t_k^{signal} &= 0 && (M < cycle_k \llcorner split_k^{green}) \\
 &cycle_k \llcorner split_k^{red} - (M - cycle_k) \llcorner split_k^{green} && (M \geq cycle_k \llcorner split_k^{green}) \\
 M &= \mathbf{mod}((T_k - T_0 - offset_k)/cycle_k)
 \end{aligned} \tag{5}$$

M is a variable which judges the situation of traffic signal when a vehicle arrives at intersection. When the signal state is judged as “green” at the arrival time the waiting time generated by the traffic signal at intersection is 0. On the other hand, when the traffic signal indicates “red” at the arrival time, the vehicle must wait for changing to the next “green” signal. In Expression (5), $cycle_k$, $split_k^{red}$, $split_k^{green}$ and $offset_k$ are the cycle length, ratio of “red” signal, ratio of “green” signal time, offset for the signal $signal_k$ corresponding to entrance node n_i^l connected to the internal link $l_{k,i}^{outside}$. However, the value is very changeable within one signal cycle because the time cost for this evaluation point is dependent on the timing of signal switching. If the prediction errors of arrival time at intersection were happened, the possibility that the time cost is far from the originally set cost is too high. Thus, we estimate the timing for cost update and route alteration after the vehicle has passed through intersection.

t_k^Q is the running time of internal link, which contained the stop time and delayed time, occurred by other vehicles at intersection. In our case, +-intersection (cross-type

of intersection) is assumed: one entrance node is connected to three types of internal links such straight, right-turn and left-turn. Even if internal link connected to the same entrance node, the time cost after having entered the intersection must be computed corresponding to each case. Usually, when the vehicle passes through intersection, the speed is controlled so as to be slow. Thus, since the speed, measured in road segment, is not kept as it is in the intersection, the average running time t_i^0 in internal link is set usually as the attribute of internal link. We make use of VDF(Volume Delay Function) [7] with a view to computing running time cost in internal link. The feature in VDF is shown in Figure 9: VDF represents that the running time is delayed according to the increase of traffics on road segments. Also, the formula is as follows:

$$t(v) = t_0 \parallel VDF(V/C) \tag{6}$$

t_0 : running time cost in free flow,
 V : traffics,
 C : traffic capacity.

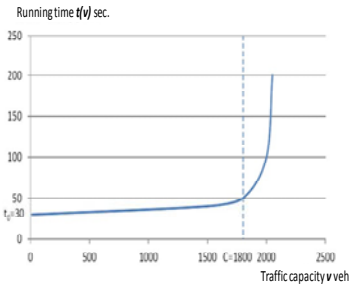


Fig. 9. VDF(capacity=1,800; average running time:30)



Fig. 10. Map used in experiment

Though we can understand the characteristics in Fig.9, the running time cost $t(v)$ extends very slowly from the running time cost t_0 ($=30$) in free flow, according to the amount of traffics. The traffic volume V does not override traffic capacity C ($=1800$); and when C is 1800 or more the running time cost drastically increases. In our case, we disregard the very large value of V/C because the traffic situation to be investigated is not crowded specially. Though the formula such as BPR [8], Conical [9], etc. have been proposed until today as an expression of VDF, we use BPR(Bureau of Public Roads) which is popular in various domains to compute the running time cost of internal link.

$$f^{BPR}(x) = 1+x^\alpha \tag{7}$$

Here, the parameter α is a variable to specify the application range and variance width. Using these means, we compute the capacity Q at the entrance node of the intersection, and store it as the attribute in the entrance node. In this case, the update cost of capacity is high since the variables relate to the estimation of capacity Q . Also, the capacity Q has each entrance node as constant values since Q related to a part of internal link cost, but not to all. Thus, we compute the link cost using VDF, looking upon the capacity Q in the entrance node as the traffic capacity for VDF. On the other hand, the number of vehicles which can pass currently within 1 cycle in order to extend link, which is connected to the entrance node, is regarded as the traffic volume for VDF.

t_k^Q is represented as follows:

$$t_k^Q = t_k^{inside} \lceil VDF(F/Q_k) \rceil \tag{8}$$

Here, t_k^{inside} is an average running time of the corresponding internal link l_k^{inside} , F_i is the number of vehicles which can pass within the last 1 cycle to the external link $l_i^{outside}$ connected to the entrance node n_k^l .

4 Experiment and Evaluation

Our experiments are performed to make our method effective and prove that our method could select the cost-effective vehicle route from start point to end point in city environment.

4.1 Simulation Environment

We made use of the traffic simulator ‘‘Aimsun6’’ with a view to constructing a road network and computing the cost-effective route. The road network was constructed in some wards in Nagoya City, as shown in Figure 10. Also, Fig.10 is interpretatively transformed into the road network in Figure 11. Our road network has the following features.

- Number of nodes: 1,864 (no. of entrance nodes: 971, no.of exit nodes: 893)
- Number of links: 1,635 (no. of external links: 483, no.of internal links: 1,152)
- Practical range: 14.00 km \times 12.00 km
- Number of intersections: 543

We consider three densities of crowded vehicles, and prepared our experiments under three situations:

- 1) Low: vehicles can move in legal speed without delay;
- 2) Normal: vehicles do not wait in long queue for a long time but cannot run in legal speed;
- 3) High: many vehicles must wait for a long time because of their delay.

Here, we compare the shortest route searching, based on the existing distance and traffics with our proposed searching, included the running time cost on insides of

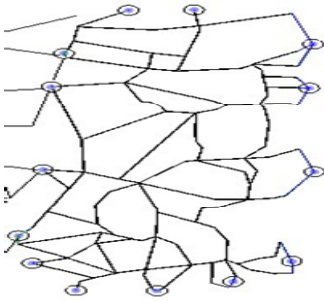


Fig. 11. Road network

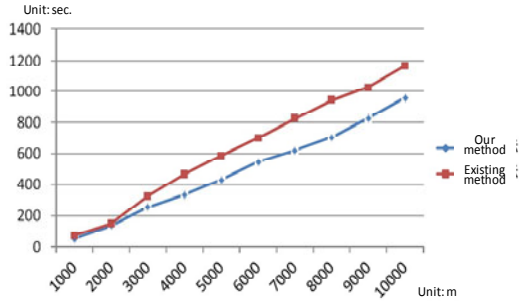


Fig. 12. Simulation in low state

intersection. These two searching methods were implemented as searching algorithms, after having applied Python script on simulator and composed the road network in Java.

4.2 Experimental Results

Figure 12, Figure 13 and Figure 14 show individual results, corresponding to the previous three situations. In the first case that the density of vehicles is low, our searching method is superior to the existing searching method, as shown in Fig.12. The longer the running route is, the more effective our searching method is because the route contains a number of intersections according to the length. In the second case that the density of vehicles is normal (or average), our searching method is superior to the existing searching method, but the comparative effectiveness is limited in 4.00 km: the difference from the first case effect is not sufficiently observed even if the total length of route become long, as shown in Fig.13. Finally, the third case

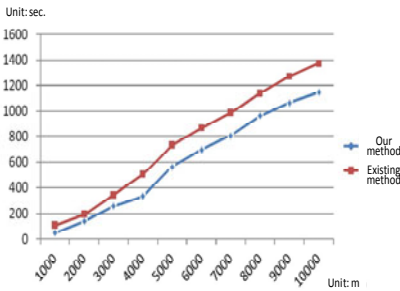


Fig. 13. Simulation in normal state

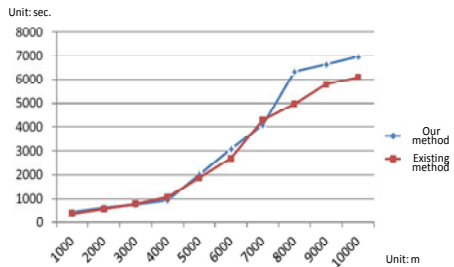


Fig. 14. Simulation in high state

that the density of vehicles is high is shown in Fig.14. Fig.14 does not show that our searching method is more effective than the existing method. In particular, the longer the length of running route becomes the less the effectiveness of our searching method is.

5 Conclusion

In this paper we proposed an advanced searching method of shortest route, which take account into the waiting time occurred by traffic signals at intersections. In order to attain this searching method practically, we introduced a road network model which can look upon the waiting time of traffic signals at intersections as the cost: this model specifies routes in insides of intersections as internal links, and regards road segments between intersections as external links. Also, we made the recommendation method of effective route searching clear after having computed the cost of each link based on switching time of traffic signals and traffic situation, and then estimated the optimal route by Dijkstra algorithm. From a viewpoint of our experiment and evaluation, our searching method makes it possible to reduce the time cost of about 20% or more at maximum in comparison with the existing method. As one of our future work, it is necessary to investigate route searching algorithm when the difference between estimated time and the practical derived time is too large.

References

1. Ezawa, H., Mukai, N.: Adaptive traffic signal control based on vehicle route sharing by wireless communication. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6279, pp. 280–289. Springer, Heidelberg (2010)
2. Bazzan, A., Boffo, F., Klugl, F.: Avoiding the Braess Paradox with Information Manipulation. In: Proc. of Workshop on Agents in Traffic and Transportation in 3rd International Joint Conference on Autonomous Agents and Multi-agent Systems, pp. 1–7 (2004)
3. Yamashita, T., Izumi, K., Kurumatani, K., Nakashima, H.: Smooth Traffic Flow with a Cooperative Car Navigation System. In: Proc. of International Joint Conference on Autonomous Agents and Multi-agent Systems, pp. 478–485 (2005)
4. Bruce, G.: Shortest-path Algorithm: A Comparison. *Operations Research* 24(6) (1976)
5. Rice, J., Zwet, E.V.: A Simple and Effective Method for Prediction Travel Times on Freeways. *IEEE Trans. on Intelligent Transportation System* 5(2), 200–207 (2004)
6. Yu, J., Chang, G.-L., Ho, H., Liu, Y.: Variation Based Online Travel Time Prediction Using Clustered Neural Networks. In: Proc. of International IEEE Conference on Intelligent Transportation Systems, pp. 85–90 (2008)
7. Jastrzebski, W.P.: Volume Delay Function. In: Proc. of International EMME/2 Users' Group Conference, pp. 1–7 (2000)
8. Bureau of Public Roads, Traffic Assignment Manual, U.S. Department of Commerce (1964)
9. Spiess, H.: Conical Volume Delay Functions. *Transportation Science* 24(2) (2000)

Virtual Organization Networking Strategies – Simulations Experiments

Konrad Fuks and Arkadiusz Kawa

Poznań University of Economics,
al. Niepodległości 10, 61-875 Poznań, Poland
{konrad.fuks, arkadiusz.kawa}@ue.poznan.pl

Abstract. In recent years, all kinds of networks, especially social and business ones, have been developing very dynamically. The increase in their importance is mainly caused by the popularization of information technology [2]. The Internet, as a means of communication and a platform of business interactions, has caused commercial cooperation and exchange to get a global dimension. The participants of networks share their resources and carry out investments (e.g. technology development, market research, launching new products), which may improve the competitiveness of individual parties as well as the whole network. The main contribution of this paper is the presentation of networking strategies representing different approach to business cooperation. The authors have built and verified a simulation model. The results of the simulation experiments have allowed to analyze the structures of the networks which depend on networking strategies.

Keywords: agent based simulation, network economy, networking, virtual organizations, NetLogo.

1 Introduction

If we think about the surrounding reality, we can undoubtedly state that we live in a highly networkized world. Everything around us and we ourselves are a set of various networks. The human being is built of biological networks (e.g. bloodstream or brain) and functions within specific food chains (our food consists of many different elements) which, from the human perspective, are nothing else but a network. We are surrounded by networks of gravitational connections of stars, planets and galaxies constituting an unexplored network – the universe. When we look at interpersonal relations and the business interactions resulting from them, we can also see networks. Our reality, from the point of view of the human civilization, is a set of social, political, cultural, technological and, obviously, business networks [3]. The last ones have recently gained special significance. They are based on economic transactions carried out among enterprises.

Economic networks mainly consist in increasing flexibility and developing enterprises' ability to adapt. The globalization of markets and competition imposes more and more requirements on these two attributes of enterprises. Customers expect highly individualized products, intense competition demands constant innovation and keeping pace with the dynamically changing environment. An enterprise functioning in isolation is not able to cope with all the above.

The structure of the paper is as follows: the characteristic of virtual organizations is discussed in Sec. 2. Then, Sec. 3 describes properties of networks and Sec. 4 presents networking strategies and simulation assumptions proposed by the authors. Next, Sec. 5 contains the results of the simulation experiments. The conclusions and directions for future research are outlined in Sec. 6.

1.1 Virtual organizations

The shift of the axis of the world economy function towards knowledge, unique resources and outsourcing being in progress for several years has caused new organizational models to emerge. They are based not only on the aforementioned networks, but also on virtuality.

Virtuality, with reference to enterprises, is understood as the use of the network character of business relations in response to opportunities occurring on the markets. A virtual organization was first defined in 1986 by Mowshowitz in his article “Social dimensions of office automation” [8]. The following years brought an increase in the number of both the researchers interested in the subject and the definitions of a virtual organization. The elements most frequently taken into consideration when defining a virtual organization are the following [10]:

- The significance of information technology and information resources in the organization.
- The type of goals which the organization should strive to achieve.
- The dominant features of the relations of the organization with the environment.

An enterprise undergoing the process of virtualization oversteps its boundaries in search of potential partners with whom it may establish cooperation in order to achieve a specific benefit. The partners do not only play the traditional role of suppliers, but also build a value chain together (a virtual organization) which creates a product consistent with the final customer’s needs. The virtualization of an organization based on relations and interdependence is becoming a social phenomenon with very real consequences [4]. Virtuality as a paradigm of the study of management is a response to the following trends [7,12]:

- More and more often, products and services are based on information and processes of knowledge management, thanks to which they may be created, popularized and sold in a digital form (virtual product).
- The era of the Internet has redefined the way of doing business, making it possible to work away from the office at relatively low costs.
- Networks and mutual dependence have become common ways of organizing cooperation, ensuring an effective use of resources and a lot of flexibility as well as speed in reacting to market changes.
- Market and resources globalization has enabled companies to enter world markets, or allowed to access world-class expert knowledge, irrespective of their location.
- A new business environment requires an enterprise strategy based on the following three key elements: low costs, high quality and a fast reaction to the customers’ needs.

It is worth noting that a virtual organization consists of a network of different legally independent units which unite in order to carry out a given task. The connections between the parties creating a virtual organization are of a quasi-stable character. It results from the fact that each of the parties may resign from the participation in the virtual organization, which increases the dynamism of the organization itself at the same time [5].

2 Network Features

M.E.J. Newman described eight main features that appear to be common to networks of many different types: small-world effect, transitivity (clustering coefficient), degree distributions (scale-free networks, maximum degree), network resilience, mixing patterns, degree correlations, community structure, and network navigation [9].

In the first part of the study we decided to focus on the two most popular characteristics – small-world effect and clustering coefficient. For this reason, the rest of this section will give their broader description.

2.2 The small-world Effect

In the 1960s S. Milgram carried out several experiments in which letters passed from person to person able to reach a designated target individual in a small number of steps only – around six in the published cases [9]. This phenomenon is called “the small-world effect” and “six degrees of separation”, too [1,11].

Milgram’s experiment developed out of a question about the probability that two randomly selected people would know each other [13]. This effect has implications for a lot of processes taking place in enterprises networks. In this case most pairs of enterprises seem to be connected by a short path through the network.

The study of the small-world effect has quite a serious use: to predict the pace and scale of the spread of the epidemic, political trends, or fashions, and create programs analyzing human networks which allow you to sell more efficient products or find potential partners to do business with. For example, if one considers the spread of information about itself across a network, the small-world effect implies that that spread will be fast on most real world networks. If it takes only six steps for news to spread from any person to any other one, for instance, then the news will spread much faster than if it took a hundred steps, or even a million [9].

The shortest distance l between nodes pairs in a network is calculated from the formula (1) where d_{ij} is the geodesic distance from node i to node j , and n is the total number of nodes.

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij} \quad (1)$$

Transitivity (Cluster coefficient)

Another feature of the network is transitivity, which is often called clustering. In many networks, it turns out that if node A is connected to node B and node B to node

C, there is a high probability that node A will also be connected to node C. In other words, transitivity means the presence of an increased number of triangles in three sets of network nodes, each of which is connected to each other. It can be quantified by determining the coefficient C in this way of grouping (2) where a “connected triple” means a single node with edges running to an unordered pair of others [9].

$$C = \frac{3 \text{ number of triangles in the network}}{\text{number of connected triples of vertices}} \quad (2)$$

4 Networking Strategies and Simulation Assumptions

In order to check the network features presented in the previous section the authors are suggesting to use the simulation model. Before this model is presented, however, let us explain what the networking strategies are.

For the experiment needs three networking strategies between nodes are distinguished. Each strategy represents different approach to business cooperation. One can pick associate partners according to the number of partner’s clients or market share, other can apply different selection criteria for example quality of product/service, city of origin or product portfolio. The main goal of the experiment is to analyze the structures of the simulated networks which depend on networking strategies. The basic networking criteria behind two of three presented strategies is number of partner’s clients i.e. number of its links.

First strategy reflects situation described by world wide web or scientific collaboration research. People tend to connect to well connected nodes. The same principle stands for web pages linkage and for citation mechanism. It’s more probable that researcher will cite papers with high citation indicator and website manager will add link to webpage with high PageRank¹. One part of business interactions tend to reflect this kind of behavior. Companies cooperate with ones having big market share, broad client base or expertise based on large number of business connections. They cooperate to adapt partner’s knowledge to their business or to achieve higher competitiveness level. All of these aspects can be represented by network of interlinked nodes and we’ve called this strategy max-networking.

Second strategy represents opposite of above one. It pertains to situation when company chooses it’s cooperates from companies that are fresh on the market (start-ups) or from companies that just have few clients. This approach can benefit networking initiator with fresh business glance. Can supply organization with innovative ideas, specialized knowledge and technology. Statistics show that majority of enterprises are micro (1-9 employees) and small (10-49 employees) [6]. When we take into consideration that each business interaction (link) requires employee attention, we can assume that average number of company’s links is rather small. So min-networking strategy can represent significant part of business cooperation.

¹ PageRank is indicator developed by the Google Internet search engine for purpose of web pages listing. The higher PageRank of a webpage has the higher probability that it will appear in the Google engine results.

However building business networks just on the basis of maximal or minimal links number can be very naïve. Companies rather use much more sophisticated and complex sets of criteria during cooperation process. However links can represent various economic aspects of business relations and can be treated as set of multiple features.

As point of reference for min- and max-networking strategies we distinguished third random-networking strategy. Although Barabasi’s research shows that business networks are far from random [2] our simple networking model is meant to show polar opposites (max, min) and transition point (random).

Evaluation of each networking strategy was simulated in the NetLogo environment. NetLogo is a programmable agent based modeling environment for simulating natural and social phenomena. Full description of environment and its capabilities and features can be found at [14].

One kind of *breed* was distinguished: nodes which represent companies. Links created after networking strategy selection reflect business connections between companies. Each node (total number of nodes is identified by *nodesNum* parameter) tries to establish finite number of links described by *linksNum* parameter of the simulation. During the simulation *linksNum* varied from 1 to 10 with step of 1 and *nodesNum* varied from 5 to 100 with step of 5. Sample interface of simulation output is presented on fig. 1.

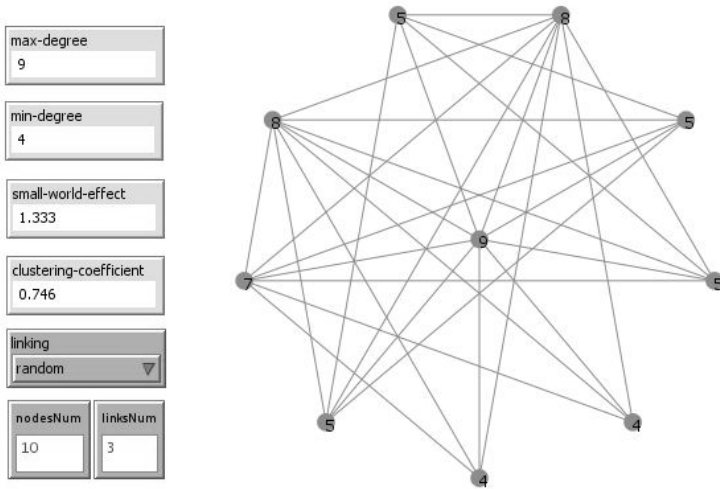


Fig. 1. Random-networking strategy – Sample simulation output

The simulation was run 100 times for each case (changing *nodesNum*, *linksNum* and networking strategy – *linking*), that gives total number of 60000 simulation experiments. Analysis of this amount of data needed statistical methods of average values estimation based on distribution of simulation output. Below section refers to average data of networking strategy simulation experiment.

5 Results of Simulation Experiments

As mentioned above, in order to check the network features (the small-world effect and cluster coefficient) three networking strategies have been used.

The calculation of the small-world effect (3) is consistent with slightly modified formula (1) of shortest geodesic path in small world model presented in [9], where d_{ij} is distance between any two nodes in the network. As there is no self linking possibility each analyzed network has possible $n(n-1)$ connections between any two nodes where n is total number of nodes.

$$l = \frac{1}{n(n-1)} \sum_{i \geq j} d_{ij} \tag{3}$$

Figure 2 presents distribution of small-world effect according to nodes (*nodesNum* – horizontal axis) and links (*linksNum* – vertical axis) numbers when min-networking strategy was selected. Distribution graph of small-world effect in random-networking strategy looks almost equally although there is smaller dispersion of values. In the case of max-networking strategy we’ve observed homogenous distribution of effect. The maximal and minimal values of l are presented in table 1.

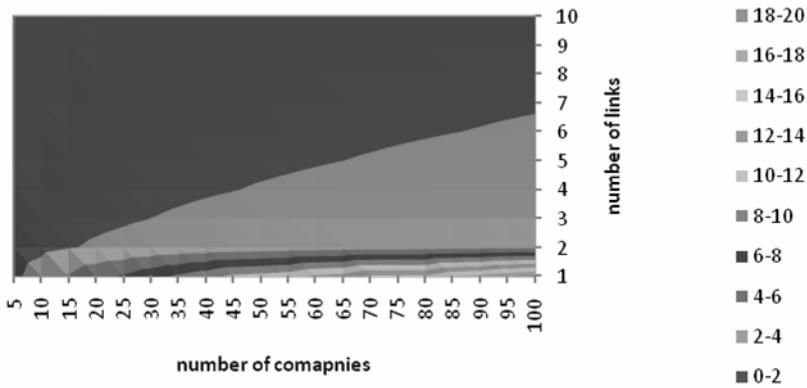


Fig. 2. Min-networking strategy, Small-world effect

Table 1. Small-world effect values dispersion

	max-net. strategy	min-net. strategy	random-net. strategy
maximal value	1,98	18,57	11,16
minimal value	1,00	1,00	1,00

Other network property is cluster coefficient value which was described in sec. 3. As there is no unified equation for its calculation we use for our experiment one implemented by Uri Wilensky in his Small Worlds model [15]. He defines cluster coefficient as: “The clustering coefficient is a measure of this “all-my-friends-know-each-other” property. This is sometimes described as the friends of my friends are my

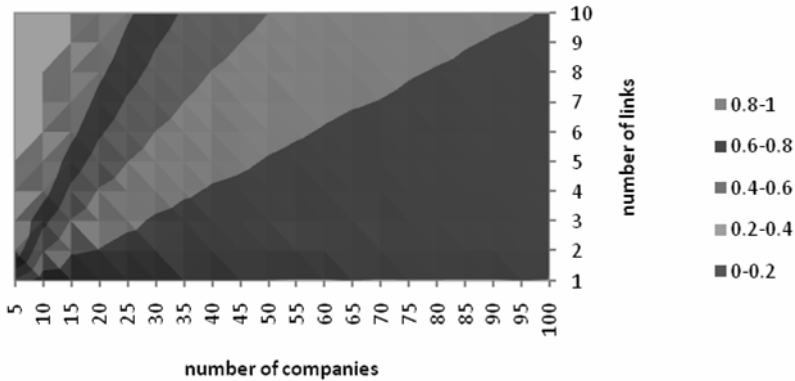


Fig. 3. Random-networking strategy, Clustering coefficient

friends. More precisely, the clustering coefficient of a node is the ratio of existing links connecting a node's neighbors to each other to the maximum possible number of such links. The clustering coefficient for the entire network is the average of the clustering coefficients of all the nodes.” [15]

Simulation output shows that in max-networking strategy cluster coefficient value is very high and in almost 90% of cases exceeds 0,9. This means if node A is connected with node B and B is connected with node C that probability that node A is connected with C is above 90%. Random- and min-networking strategy show much more disperse view of clustering coefficient and one of them is presented on fig. 3.

6 Conclusions and Future Work

A virtual organization must react fast to the changes in the environment, related both to demand and supply. Building a virtual organization on the basis of key competences enables to adjust the structure and possibilities of the organization to the changing environment in a faster and more precise way. In addition, business partners participate in the accomplishment of the aim; therefore, abilities, knowledge, costs, access to markets and the risk of failure are shared.

The simulation experiments confirmed the observation [2,3,9] that virtual organizations, like people, tend to link with well connected nodes. As mentioned, companies want to collaborate with ones having big market share, a broad client base or expertise based on a large number of business connections. But keep in mind that most enterprises protect their customers, experience, technology at all costs. If they see that their competitors are doing something similar, they probably will not work with them. They do not perceive well-networked nodes as an opportunity, but rather as a threat. Of course this is changing, but very slowly. The results of our simulation show that companies should choose a max-networking strategy which gives them best conditions of cooperation. The basis, however, is trust. There is no question of opportunism.

The future work is to check the other popular network features (degree distributions, network resilience, mixing patterns, degree correlations, community structure, and network navigation) in relation to the three networking strategies.

The authors also want to obtain data on the cooperation between enterprises on the Polish market in order to verify the simulation model and find different dependences in networks.

References

1. Almaas, E., Kulkarni, R.V., Stroud, D.: Characterizing the structure of small-world networks. *Physical Review Letters* 88 (2002)
2. Barabasi, A.L.: *Linked: How Everything Is Connected to Everything Else and What It Means*. Penguin Group (2003)
3. Barabasi, A.L.: Emergence of Scaling in Random Networks. *Science* 286 (October 15, 1999)
4. Berger, P.L., Luckmann, T.: *The social construction of reality*. Anchor Books, NY (1967)
5. Bultje, R., Wijk van, J.: Taxonomy of Virtual Organisations, based on definitions, characteristics and typology. *VONet Newspetter* 2(3) (1998)
6. <http://www.oecd.org>
7. Keinanen, K., Inas-Kukkonen, H.: Virtual Organizing as a Strategic Approach to Stay Competitive – A Conceptual Analysis and Case Study. In: Malhotra, Y. (ed.) *Knowledge Management and Virtual Organizations*. Idea Group Publishing, Harshey (2000)
8. Mowshowitz, A.: Social dimensions of office automation. *Advances in Computers*, 25
9. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45 (2003)
10. Niemczyk, O., Olejczyk, K.: Organizacja wirtualna (Virtual organization). In: Krupski, R. (ed.) *Zarządzanie Przedsiębiorstwem w Turbulentnym Otoczeniu (Enterprise management in a turbulent environment)*. PWE, Warsaw (2005)
11. Schnettler, S.: A small world on feet of clay? A comparison of empirical small-world studies against best-practice criteria. *Social Networks* 31(3) (2009)
12. Skyrme, D.J.: *The Realities of Virtuality*. In: *Organizational Virtualness Proceedings of the First VONet – Workshop*. Simova Verlag, Bern (1998)
13. Travers, J., Milgram, S.: An Experimental Study of the Small World Problem. *Sociometry* 32(4) (1969)
14. Wilensky, U.: *NetLogo itself*. NetLogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University. Evanston, <http://ccl.northwestern.edu/netlogo/>
15. Wilensky, U.: *NetLogo Small Worlds model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston (2005), <http://ccl.northwestern.edu/netlogo/models/SmallWorlds>

Multistage Covariance Approach to Measure the Randomness in Financial Time Series Analysis

Ryszard Szupiluk^{1,2}, Piotr Wojewnik^{1,2}, and Tomasz Ząbkowski^{1,3}

¹ Polska Telefonia Cyfrowa Ltd, Al. Jerozolimskie 181, 02-222 Warsaw, Poland

² Warsaw School of Economics, Al. Niepodległości 162, 02-554 Warsaw, Poland

³ Warsaw University of Life Sciences, ul. Nowoursynowska 159, 02-787 Warsaw, Poland
{rszupiluk, pwojewnik, tzabkowski}@era.pl

Abstract. The paper presents a new method for randomness assessment in data with temporal structure. In this approach we perform multistage covariance analysis on several parts of the signal to synthesize information about variability and internal dependencies included in its structure. This allows us to identify deterministic cycles or to detect the level of randomness in signals what is an important issue for the design of transactional, prediction and filtration systems. To confirm validity of the proposed method we tested it on simulated and real financial time series.

Keywords: information representation, variability measure, volatility, modeling financial data, noise detection.

1 Introduction

The unstable nature of modern financial markets and the models that operate on transactional systems require precise information on the variability and dependencies present in financial instruments [8]. At the same time more and more transactions in today's financial markets are performed by trading system which trades on investor account automatically [10]. This means that the financial data reflect not only the human perception, aspirations and emotions but also automated strategies which are able to adapt itself according to the market conditions. Due to the nature of the modern financial markets, the models that describe them require precise information about the variability and the dependencies that are present in a particular time series such as stock indexes, foreign exchange rates or commodity prices. This changes the nature of the market by increasing the variety of possible mathematical characteristics to analyze [5, 6, 16]. There are many approaches to capture the changes in data and depends on the context and type of problem we can say about the variability, fluctuations, dispersion, volatility or smoothness of the time series [4, 19, 20]. For simplification we assume that, in general, we are investigating a feature called variability. Taking into account the possible relationships that exist between the values of a time series we may say about the internal dependencies, correlation, autocorrelation, memory effects, associations or cross interactions. From this perspective, we label these phenomena with the term "internal dependencies". One of

the reasons to develop these techniques is the need to assess the level of signal randomness for such tasks like prediction, filtration, pattern recognition or separation [2, 17, 18]. Therefore, this publication presents a new method that aims to measure or to assess the level of signal randomness in financial time series or signals. In this method, we combine the properties of the correlation analysis with the structural approach of rescaled range R/S analysis [9, 13]. In these considerations, we will focus on data, in which the data sequence is important – what is known as time series, signals or data with the temporal structure.

This concept seems to be particularly important in modeling behavior of financial markets as well as in formulating the investment systems. In particular, it can be applied to the automated trading systems which are based on artificial intelligence, neural networks, agent and multi-agent systems, and basically everywhere where the information representation plays an important role.

2 Noise Detection Problem

The term random signal or random noise means a lack of knowledge about the phenomenon which generates the data or its mathematical model is not known [19]. In particular, this means there are no clear patterns and relationships in empirical data and we may presume that data with random noise should be irregular and mutually independent. In this way, the concepts such as randomness, variability, and dependency somehow intertwine with each other. From analytical perspective the most popular approach that joins those concepts is Second Order Statistics (SOS) in which the variability is associated with the variance as the measure of square dispersion around the average. The SOS approach has strong theoretical foundation in Gaussian distribution properties and the Central Limit Theorem – from which we can conclude that Gaussianity is the essential for the randomness. In practice, it is very convenient that the uncorrelated Gaussian variables are equivalent to statistically independent variables. If the mean value and the variance for the Gaussian distribution are known then we can identify it, which means that we have complete statistical information about the analyzed data. The abovementioned SOS properties and advantages are the foundation of many filtration, identification and data separation techniques [1, 4, 7, 11, 19].

However, there are number of problems where the application of this approach encounters significant difficulties. Primarily, we may raise the issue associated with the existence of second-order statistics for data with alpha stable distributions ($\alpha < 2$), which occur relatively frequently on the financial markets [12, 13, 14]. Secondly, the other problems are associated with the proper description and identification of long memory effects, volatility segmentation, discovering complex patterns of mutual dependencies, which in fact often concerns the issues associated with high-density data [5, 13]. Thirdly, the basic limitation of variance as a single parameter to measure the variability is the lack of information exploration on the temporal structure in data. Obviously, the temporal structure of the data can be explored by autocorrelation function but the function requires the assumption about the stochastic process, which often opens the discussion whether the adopted model

actually describes the analyzed problem. For example, is it reasonable to adopt the explanation that IBM stock price is one of the possible implementation of "some" general stochastic process? What is more, in terms of stochastic processes we often face the difficulties to verify the fundamental assumptions about the stationarity and the ergodicity [14, 19]. Even if we cope with those assumptions we face the issue of how to compare the autocorrelation functions and then how to aggregate them to single parameter that is useful for algorithmic trading system. Other well-known autocorrelation function limitations concern the low ability to capture the long memory effect in data due to its exponential decline. These disadvantages draw the interest towards the new techniques what enabled to develop such analysis methods as R/S (rescaled range) and Hurst exponent analysis, the Stone's measure of predictability or the methods associated with the fractal dimension [3, 9, 15]. In particular, this involves the Peters work regarding the application of Hurst exponent on financial markets [13]. One of the reasons of Hurst's exponent popularity is its relationship with the fractal analysis (fractal dimension), alpha stable distributions and fractional Brown's motion [13, 14].

The Hurst exponent can be obtained from R/S analysis. For analyzed signal y we perform splits into parts with n - observation each and calculate as

$$E\left\{\frac{R}{\sigma}\right\}_n = cn^H. \quad (1)$$

where $R = \max(y) - \min(y)$, and σ is standard deviation, n is the number of observation in each part, expectation is taken over the all parts. We can obtain H taking the logarithm of (1) and perform regression on n , where c is some constant.

The Hurst exponent can take values form 0 to 1, where for $H=0.5$ we have white noise (not necessarily Gaussian) and for $H>0.5$ we have persistent signal (e.g. $H=1$ means pure deterministic signal) and $H<0.5$ we have antipersistent signal. It is important to calculate H only on linear part of the regression identified by individual inspection during analysis. One of the advantages of the Hurst's exponent is its possibility to explore the temporal structure in time series. The other interesting feature of the R/S is structural approach, in which final H value is obtained from aggregation of information from several partitioning stages [9, 13].

However, to estimate precisely the value of the Hurst's exponent we need to analyze each problem individually, since the value of the exponent must be determined on the basis of the linear regression dependence. In addition, the Hurst's exponent is sensitive to extreme values. Therefore, the R/S method can be quite effective for very detailed expert research and rather inadequate for automatic algorithms in noise detection, filtration or separation problems.

3 Novel Variability Measure

In this paragraph we present a new method in which we try to combine the ideas of the correlation analysis with the concepts of structural analysis of R/S. We expect that this new measure will meet the following requirements:

- a) based on Second Order Statistics – what gives strong theoretical background that is computationally simple,
- b) able to synthesize information that come from different structural parts of the signal,
- c) can capture the effect of long memory,
- d) robust to outliers,
- e) uses the information that is inherent in the temporal structure.

Our considerations will be based on the variance as a measure of volatility and covariance as a measure of dependence. The variance can be noted as $\text{cov}(\mathbf{x}_i, \mathbf{x}_j) = E[(\mathbf{x}_i - E[\mathbf{x}_i])(\mathbf{x}_j - E[\mathbf{x}_j])] = c_{x_i x_j}$, the covariance can be noted as $\text{var}(\mathbf{z}) = D^2(\mathbf{z}) = E[(\mathbf{z} - E[\mathbf{z}])^2] = \sigma_z^2$. At the same time we will explore the fact that these concepts are interrelated. This relationship is especially apparent when we deal with multidimensional analysis. The application of multidimensional analysis for a single time series \mathbf{z} means that we need to obtain a set of variables from it. We can do this in several ways, e.g. to perform decomposition on the hidden components $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ to obtain $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$ or to divide into parts $\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \mathbf{x}_3^{(n)}$ with n – observations to obtain $\mathbf{z} = [\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \mathbf{x}_3^{(n)}]$ (block matrix notation).

To present the second order statistics relationship between the variability and dependence we start with a simple case of variable \mathbf{z} decomposition into three components $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ (e.g. a typical decomposition into trend, cyclical and random components). In this way we have $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$ and for variance $\text{var}(\mathbf{z}) = \text{var}(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)$ we have:

$$\text{var}(\mathbf{z}) = \overbrace{\text{var}(\mathbf{x}_1) + \text{var}(\mathbf{x}_2) + \text{var}(\mathbf{x}_3)}^{\text{variability}} + \overbrace{2\text{cov}(\mathbf{x}_1, \mathbf{x}_2) + 2\text{cov}(\mathbf{x}_2, \mathbf{x}_3) + 2\text{cov}(\mathbf{x}_1, \mathbf{x}_3)}^{\text{dependence}} \tag{2}$$

As we can see in (2) the variance of the signal \mathbf{z} is the sum of the variability of individual components and their dependencies. In general, we can write

$$\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_m \text{ as}$$

$$\text{var}(\mathbf{z}) = \text{var}(\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_m) = \overbrace{\sum_{i=1}^m \text{var}(\mathbf{x}_i)}^{\text{variability}} + 2 \overbrace{\sum_{\substack{i=1, j=1 \\ i \neq j}}^m \text{cov}(\mathbf{x}_i, \mathbf{x}_j)}^{\text{dependence}} = \sum_{i=1}^m \sigma_i^2 + \sum_{\substack{i=1, j=1 \\ i \neq j}}^m c_{ij} \tag{3}$$

The expression (3) can be reduced to the analysis of variance and covariance for individual components which actually brings the problem to the analysis of the covariance matrix \mathbf{C}

$$\mathbf{C} = E[\mathbf{xx}^T] = \begin{bmatrix} \sigma_1^2 & c_{12} & \dots & c_{1m} \\ c_{21} & \sigma_2^2 & \dots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & \sigma_m^2 \end{bmatrix} \tag{4}$$

If we take $c_{ii} = \sigma_i^2$ the expression (3) can be taken as

$$\text{var}(\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_m) = \sum_{i,j=1}^m c_{ij} = \text{trace}(\mathbf{C}) + \sum \text{off}(\mathbf{C}) \tag{5}$$

where $\sum \text{off}(\mathbf{C})$ means the sum of matrix \mathbf{C} elements outside the main diagonal, $\text{trace}(\mathbf{C})$ means the sum of matrix \mathbf{C} elements on the diagonal.

In further considerations we apply the relations (2) - (4) between variability and dependence not for latent components but for chosen parts of the original signal. We split original signal into parts with equal number of observations several times. This means that in the first stage we would split our signal \mathbf{z} on m_1 intervals with the same number of n_1 observations and then to calculate the covariance matrix \mathbf{C}_{n_1} . In the second stage we split our signal \mathbf{z} on m_2 intervals with the same number of n_2 observations and this time we calculate the covariance matrix \mathbf{C}_{n_2} for the extracted parts \mathbf{x}_i , $i = 1, \dots, m_2$. In next stages we proceed analogously for n_i, m_i . Note that taking N observations in the original signal we would have $N = m_i n_i$. As the result we obtain a set of covariance matrices \mathbf{C}_n for the subsequent splits with a given number of observations n . Fig. 1 presents the abovementioned idea.

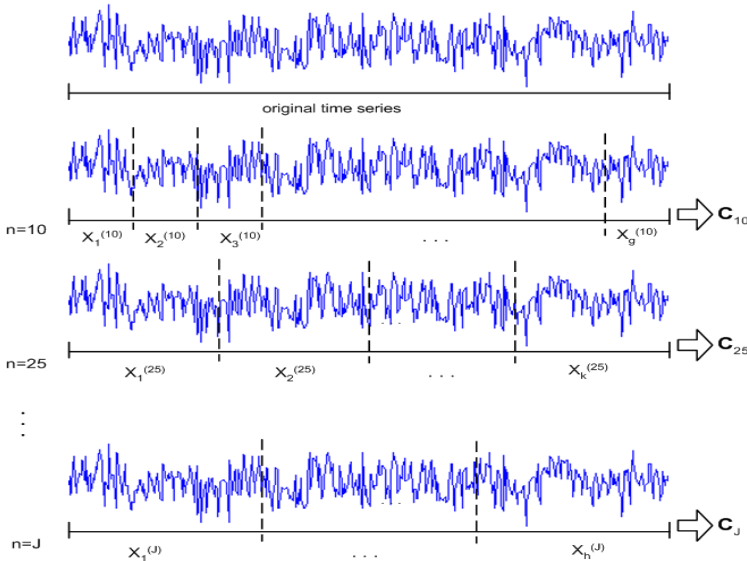


Fig. 1. The concept for time series analysis with a covariance matrices \mathbf{C}_n obtained for the parts with a different number of observations

The analysis of the matrix C_n is directed to look for interesting relationships between variances and covariances what can be written as pattern or relation recognition (we name it as ξ - ksi) between the matrix elements C_n

$$\{\sigma_1^2, \sigma_1^2, \dots, \sigma_m^2\}_n \xrightarrow{\xi \text{ relation}} \{c_{12}, \dots, c_{1m}, c_{21}, \dots, c_{2m}, \dots, c_{m1}, \dots, c_{mm-1}\}_n \quad (6)$$

As relatively simple and intuitive indicator we take the ratio of non-diagonal (covariance based) to the diagonal elements (variance based) such as:

$$\bar{\xi}_i(n) = \frac{1}{(m-1)} \frac{\sum \text{off} C_n}{\text{trace}(C_n)} \quad (7)$$

where m is the number of separate parts extracted from the original time series, n is the number of observations, and i denotes the extraction into i -th part. This dependence (7) has a value of 0 for the lack of relationships between different parts of the signal parts and the value of 1 if the correlation is maximal. Since we calculate the dependence (7) for the different number of parts with different n we get the set of values (7). This set characterizes the structural dependence and variability in the analyzed signal. The information synthesis may be given by the regression $\bar{\xi}$ of n or may be inspected as an average:

$$\bar{\xi} = \frac{1}{K} \sum_{i=1}^K \bar{\xi}_i \quad (8)$$

where K is the number of $\bar{\xi}_i$. The advantage of (8) is the ability to distinguish quickly the random signals from deterministic ones (in terms of variability and inter-internal dependency).

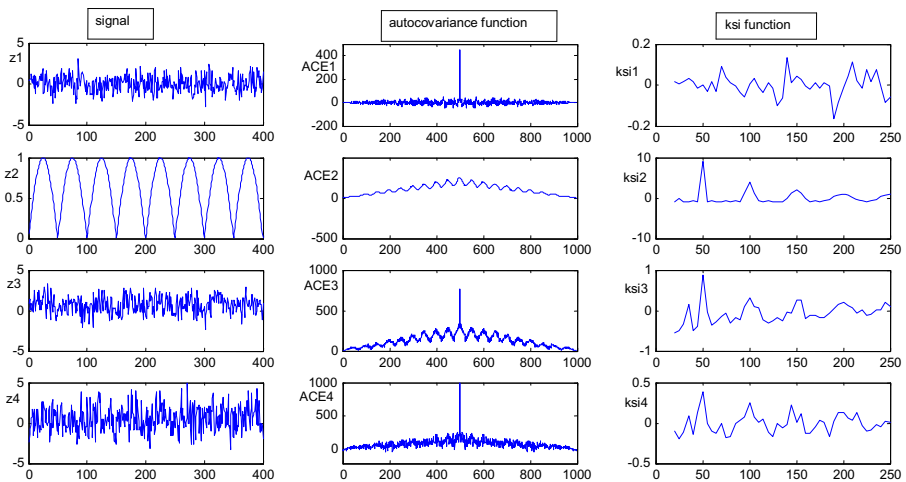


Fig. 2. The detection of cycles with regression $\bar{\xi}$ on n . First column of the figure presents the signals for z1: random noise $N(0,1)$, z2: $\text{sign}(\sin(2\pi/100*t))$, z3, z4 are z2 hidden in noise with SNR factor -9dB and -13dB, respectively. Second column presents autocorrelation functions and third one presents ksi factors. There, we can see that the values for the period number 50 reach a local minimum.

It allows also analyzing the presence of cycles in the data as presented in Figure 2 where we present signals with different SNR - where SNR means signal to noise $SNR = 10 \log_{10}(\text{var}(\text{signal}) / \text{var}(\text{noise}))$. Unfortunately, the aggregated dependence (8) has a low ability to classify the signals with a similar level of noise.

To obtain a far more sensitive measure to the differences in the level of noise we replaced the standard summation of the matrix \mathbf{C} elements with the new norm calculated on the matrix elements. This would yield the following expression:

$$\Xi(n) = \frac{1}{(m-1)} \left(\frac{\|\mathbf{C}_n\|_p}{\|\text{diag}(\mathbf{C}_n)\|_p} - 1 \right). \tag{9}$$

where $\text{diag}(\mathbf{C}_n)$ is diagonal matrix with entries c_{ii} , $\|\mathbf{C}\|_p = \left(\sum_{i=1}^m \sum_{j=1}^m |c_{ij}|^p \right)^{1/p}$ is

the p -norm with the most popular cases like 1-norm is absolute value criterion, 2-norm is Frobenius norm, ∞ -norm is Chebyshev norm. The final measure is obtained as the mean value (9) for different n

$$\Xi = \frac{1}{K} \sum_{i=1}^K \Xi_i. \tag{10}$$

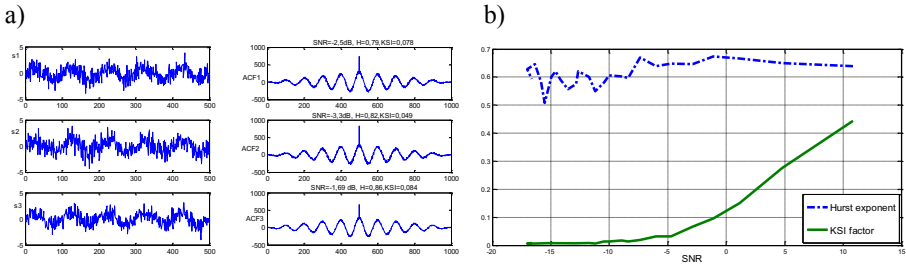


Fig. 3. a) The sinusoids with noise signals and its measures for randomness estimation using autocorrelation function, Hurst exponent and the KSI factor; b) The relationship between the KSI and Hurst exponent values for a sinusoid with different levels of signal to noise ratio

The outcome is the measure of randomness level in a time series. The Figure 3 presents the comparison of randomness using autocorrelation function, Hurst exponent and the KSI Ξ factor. The last one, that is KSI factor, is the closest to match to the true randomness (the level of noise).

4 The Algorithm Outline to Measure the Randomness

The algorithm to measure the randomness (the level of noise) in signals can be written as:

1. Split the signal x into L parts of n observation each to have a set of signals $\mathbf{x}_i^{(n)}$, $i = 1, \dots, m$.
2. Store the signals in one multivariate variable

$$\mathbf{x}_n = [\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \dots, \mathbf{x}_m^{(n)}]. \tag{11}$$

3. Find the covariance matrix $\mathbf{C}_n = E[\mathbf{x}_n \mathbf{x}_n^T]$ and compute

$$\Xi(n) = \frac{1}{(m-1)} \left(\frac{\|\mathbf{C}_n\|_p}{\|diag(\mathbf{C}_n)\|_p} - 1 \right). \tag{12}$$

In the context of our considerations we set the value of p parameter to 2.

4. Make K iterations with steps 1-4 for different n to obtain set of $\Xi(n_t)$, $t = 1, \dots, K$ and perform the regression Ξ on n .
5. The simplest way to synthesize information from a set of $\Xi(n_t)$, $t = 1, \dots, K$, is the mean value calculation

$$\Xi_C = \frac{1}{K} \sum_{t=1}^K \Xi(n_t). \tag{13}$$

5 Practical Experiment

In this paragraph we apply the previously presented method for the analysis of selected financial time series. For the experiment we use the financial instruments for which we have some a priori knowledge that allow us to evaluate the results in terms of quality. Therefore, our aim is to check whether we can trust the method when assessing the level of randomness in the financial series. The main motivation of this study is to distinguish the series in which the trends and patterns are purely stochastic from the instruments which are to some extent deterministic. We should point out that the practical question is not whether the instrument is pure random but rather how much random or noisy it is. The question is what instrument to choose for the transaction, and we are more interested in assigning a specific ranking of randomness rather than absolute assessment.

A business example of deterministic situation is e.g. an attempt of hostile takeover seen as systematic purchase of shares or a conscious policy of diluting the company's shares. Such signals classification technique may be an important factor to improve the effectiveness of automated trading systems which are based on patterns exploration in the data. Given that automated systems usually search the hundreds or even thousands of instruments, it is clear that the analysis should be conducted in a completely automatic way based on the statistically reliable method.

In this experiment we analyzed the following financial instruments: foreign exchange rate for currency pair EUR/USD, Petrolinvest and Unicerdit share (from Warsaw Stock Exchange) and SP500 index (see Fig 4).

We calculated the logarithmic rate of return daily and for these time series we measured the randomness with Ξ_C . Both currencies – EUR and USD – are the basis for international transactions, speculation and reserve assets. The huge liquidity of the market, its influence on the world’s economy and the minor impact of individual players on current price cause the currency pair to perform relatively stable (in this crisis time). We can obtain the conclusion of stable result from the relatively high value of Ξ_C (EUR/USD) at the level of 0.0058.

Similarly, we can observe high value of Ξ_C for Petrolinvest stocks. The company spends a lot of money on searching for new oil field. The earnings from operational activity are not enough to finance the company expansion. Therefore the company regularly places new emissions that results in stock dilution. The up-rising moves are only incidental and do not change the general downward trend. This deterministic trend is reflected in relatively high value of Ξ_C factor at the level of 0.0064.

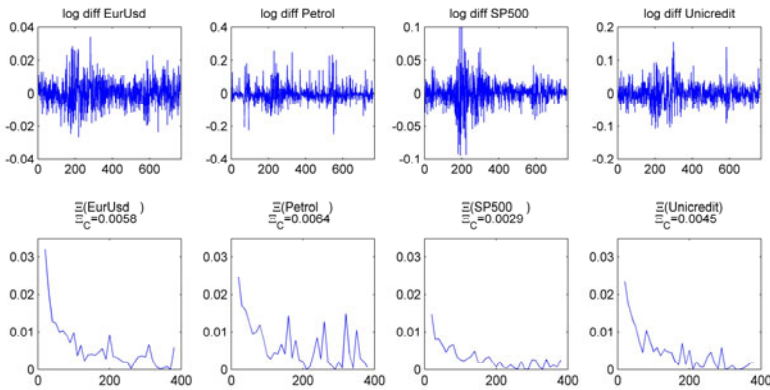


Fig. 4. First row presents logarithmic differences of EUR/USD, Petrolinvest and Unicredit stocks and SP500 index. Second row includes Ξ KSI factors for various sample size.

The time series of SP500 reflects transactions of a huge number of individual investors with various strategies, various academic background and various expectations. Moreover, most of them speculate or even invest intuitively. The behavioral aspect explains why the SP500 index performs randomly. The result is also verified by relatively small Ξ_C value of 0.0029.

Finally, the fourth instrument – Unicredit performs both features (0.0045) – deterministic – from the fundamental decisions of management and key stockholders – and random – inherited from individual investors’ behavior.

5 Conclusions

The variability and the dependency are important measures used in financial trading systems. Transactional systems require high quality information on the variability and dependencies present in financial instruments what is the motivation for development of new techniques in this area. The methods presented in this paper meet these expectations. Based on the variance and correlation concepts we developed a

multi-stage structural analysis technique that is able to assess the level of randomness, to identify the cycles in the data and to detect the long memory effects in financial time series. It is worth mentioning that the application of this method allows us to explore the temporal structure of the data without making any formal assumptions about the stochastic processes like stationarity and ergodicity. All the issues related to data scaling, standardization and initial transformations are performed as for a typical variance and correlation analysis.

This new concept has a form that is not closed and therefore it creates a wide open field for further research. The opportunity for development is related to the fact that this method is based on second-order statistics and this makes it possible to apply an elaborate apparatus of statistical inference in significance testing. The motivation for further research is also based on the experiments to be conducted on both, simulated and the real data to document the effectiveness of the method in various applications.

References

1. Brown, R.G., Hwang, P.: Introduction to random signals and applied Kalman filtering. John Wiley, New York (1992)
2. Bishop, C.M.: Neural networks for pattern recognition. Oxford Univ. Press, Oxford (1996)
3. Bunde, A., Halvin, S.: Fractals in Science. Springer, Heidelberg (1994)
4. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing. John Wiley, Chichester (2002)
5. Dunis, C.: Forecasting Financial Markets. John Wiley, West Sussex (1996)
6. Dunis, C.: Applied Quantitative Methods for Trading and Investment. John Wiley, West Sussex (2003)
7. Haykin, S.: Adaptive filter theory. Prentice-Hall, Upper Saddle River (1996)
8. Hull, J.: Options, futures, and other derivatives. Prentice-Hall, Englewood Cliffs (2000)
9. Hurst, H.E.: Long term storage capacity of reservoirs. Trans. Am. Soc. Civil Engineers 116, 770–799 (1951)
10. Kaufman, P., Hoboken, N.J.: New trading systems and methods. John Wiley and Sons, Chichester (2005)
11. MacDonough, R.N., Whalen, A.D.: Detection of signals in noise. Academic Press, San Diego (1995)
12. Nikias, C.L., Shao, M.: Signal Processing with Alpha-Stable Distributions and Applications. John Wiley&Son, West Sussex (1995)
13. Peters, E.: Fractal market analysis. John Wiley&Son, West Sussex (1996)
14. Samorodnitskij, G., Taqqu, M.: Stable non-Gaussian random processes: stochastic models with infinite variance. Chapman & Hall, NY (1994)
15. Stone, J.V.: Blind Source Separation Using Temporal Predictability. Neural Computation 13(7), 1559–1574 (2001)
16. Shiryayev, A.N.: Essentials of stochastic finance: facts, models, theory. World Scientific, Singapore (1999)
17. Szupiluk, R.: Noise Detection for Latent Component Classification in Ensemble Method. Przegląd Elektrotechniczny (1) (2010)
18. Szupiluk, R., Wojewnik, P., Ząbkowski, T.: The R/S Analysis for the Latent Component Identification for Prediction Improvement Computational Intelligence: Methods and App. EXIT, pp. 590–598 (2008)
19. Therrien, C.W.: Discrete Random Signals and Statistical Signal Processing. Prentice Hall, New Jersey (1992)
20. Vaseghi, S.V.: Advanced signal processing and digital noise reduction. John Wiley and Sons, Chichester (1997)

Multi-agent Coordination Mechanism of Virtual Supply Chain

Paulina Golinska¹ and Marcin Hajdul²

¹ Poznan University of Technology, Strzelecka 11, 60965 Poznan, Poland

² Institute of Logistics and Warehousing, Estkowskiego 6, 61755 Poznan, Poland
paulina.golinska@put.poznan.pl, marcin.hajdul@ilim.poznan.pl

Abstract. The improvement of transport processes gains a lot of interest from enterprises and governments. Configuration of supply chain should meet both economical and environmental goals. Companies participating in supply process more and more often prefer to create temporary relations and form virtual cooperation networks than to keep traditional long-term contracts. The aim of the paper is to present the potential of multiagents application for coordination of processes in virtual supply chain. The theoretical background of the project is presented. The main problems and requirements of the virtual supply chain coordination are identified. The reference model is described. Finally authors present a case study from pharmaceutical industry.

Keywords: multi-agent system, virtual organization, transport processes.

1 Virtual Supply Chain

The dynamic development of web- based services and common access to the broad band Internet enable many companies to form virtual cooperation network. Virtual supply-chain is a good example of such cooperation. Virtual supply chain can be defined as a network of individual companies that are organized around the particular customer delivery process.

The aim of the virtual supply chain is to achieve the effect of the economy of scale. It benefits from free exchange of resources and at the same time from the individual control over these resources [3]. The analyses of the previous studies in this field have showed that the virtual supply chains are described by a group of common characteristics:

- technology-application of modern IT tools and developed telecommunications infrastructure provide exchange of data necessary for the effective organization of transport processes;
- opportunistic approach to network configuration – any time companies evaluate the benefits of participation in the virtual organization and compare them with benefits of individual performance;
- temporary character of relations- usually there is no long-term contracts and companies are able the reconfigure the network by each contract;
- trust – one of the main features that allows the operation of virtual organizations, as the key to success is the openness and partnership in their activities;

- excellence – each of the members of is committed to the success of the virtual cooperation network.

Virtual supply chain can be treated as holonic structure. This results in an innovative approach to the implementation of business processes through the creation of open and reconfigurable structure able to react on changes in the economic environment [adapted from 5]. The main advantage of holonic structure is their fitness. Holonic structure consists of small and decentralized, partly independent units [3]. Moreover they are focus only on the key competences. Figure 1 presents the example of holonic structure.

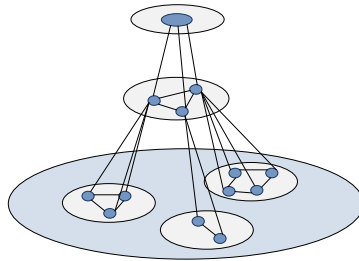


Fig. 1. Holonic structure

The essential elements of the holonic structure are so-called holons. They are cooperating nodes whose task is to collect, transform, store and share information or physical objects. Holons can work individually or cooperate forming for example multiagents system. Multiagents system allows the coordinated actions in order to solve a particular problems appearing by logistics operations. Agents can help to coordinate actions within virtual supply chain participating in search for new offers, negotiations or verifying service providers.

2 Problems of Virtual Supply Chains

Despite the fact that standards exist, industry representatives have realised that a paradigm shift in interoperability is needed. The challenge is how to combine cost efficiency with need for the reduction of the environmental impact of transport operations. An example of such industrial initiative used mainly by big companies is the Logistics Interoperability Model (LIM) by GS1[see 6]. Small and medium companies (SME) still search for the other solutions that might be suitable for them.

The importance of transport sector and its problems has made it a subject of numerous work programmes coordinated by the European Commission or by the national units. Traditionally, these projects have been quite autonomous and there has been little coordination between the projects. This situation has now changed. The partners in a number of ongoing EU funded projects (Freightwise, Euridice, Integrity, Rising, DiSCwise, SMARTFREIGHT, eFreight) start to see benefits from better cooperation.

As industrial developments and the research community have reached similar conclusions, a joint initiative has been taken to improve interoperability considerably by developing one Common Framework for exchange of information between ICT systems in transport and logistics.

Different stakeholders have their own strategies and objectives and will have to be given the possibilities to continuously develop these. As a consequence, the Common Transportation Framework needs to [4]:

- Assist in making the European transport and logistics system more efficient and environmentally friendly;
- Provide interoperability without constraining business process development and improvements;
- Lower barriers to improve business processes and develop new business models;
- Simplify interaction with authorities and compliance. For the time being there are different requirements for compliance and reporting for different modes and different geographies;
- Unlock access to services that would not be available without it;
- Enable market leaders and SMEs to interact at a low cost, and this should emerge as a standard endorsed and adopted by major freight ICT systems providers and logistics operators.

The main issues that appear by coordination of process in the virtual supply chain network are:

- trust creation among participants;
- independency from technology (in order to support the possibility of temporary relations and reconfiguration);
- interoperability and ability to facilitate interaction with existing standards (to protect investments already made in legacy and other systems).

3 Reference Model of Cooperation within Virtual Supply Chain

The main task for the reference model is to stimulate cooperation between group of manufacturing and transport enterprises in the sphere of logistics processes` implementation in supply chains. Its basic assumption is to cause positive effect on enterprise`s activity in the short and long term period (focus on load factor) and also on the region in which the enterprise operates. Cooperating entities shall exchange information electronically via dedicated electronic platforms. Figure 2 presents reference model idea, where three groups of players are defined:

- Users of logistics services – companies that are engaged in the production and/or selling of the products. These companies may have their own means of transport, logistics infrastructure or they cooperate with the providers of logistics services. They issue the demand for transportation and orders for providers of logistics services.

- Provider of logistics services – companies whose core business is the provision of logistics services. Their task is the coordination of orders issued by logistics users. Also one of the cooperating firms which has its own facilities (e.g. means of transport) and is able to provide transport services for the other transport users), then it performs also the role of transport services providers.
- Coordinator – represents the logistics users, deals with the coordination of logistics processes (e.g. analysis of the possible aggregation of the transport orders issued by the different users, price negotiations, the choice of modes of transport), cooperation with suppliers of logistics services, monitoring of the aggregated processes, the carrying out of analyses in the micro and macro. Coordinator might be financed by cooperation companies in the network, when a coordinator is their employee. The coordinator may be an independent entity supporting coordination of logistics processes, which charges the users.

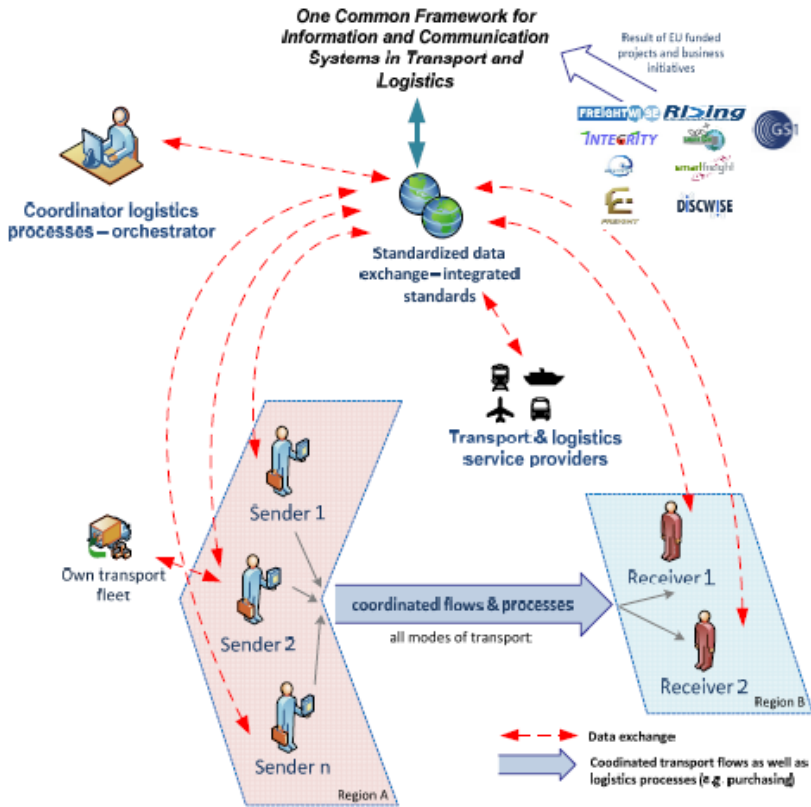


Fig. 2. Reference model idea

Developed reference model takes into consideration:

- concept of the balanced development and assumptions for the European transport policy development,
- co-modality of transport processes,
- one common standard in transport & logistics data exchange which has been developing by several EU research projects such as FREIGHTWISE, e-Freight, INTEGRITY, Smart-CM, SMARTFREIGHT, EURIDICE, RISING, DiSCwise,
- correlation between strategic planning in enterprise and current processes realization,
- correlation among logistics processes in the enterprise,
- correlation between logistics system of the enterprise and the regional/national/continental transport systems,
- requirements of enterprise's clients,
- possibility of using particular computing tools and information exchange techniques,
- particular processes' impact on effectiveness of enterprise logistics system.

Within the framework of the model the following instruments stimulating cooperation between enterprises need to be addressed during implementation phase:

- methodology enabling estimation of potential savings and profits that derive from cooperation among enterprises in logistics processes organization,
- legal framework for cooperation between shippers, logistics service providers and co-ordinator (orchestrator),
- methodology that assure common planning of logistics processes in a group of enterprises taking into consideration relation of a trade-off between transport, inventory management as well as warehousing processes,
- methodology that assures common planning of transport processes in a group of enterprises in compliance with a trade-off relation between micro scale (enterprise) and macro scale (region),
- exploitation of existing e-platforms (Kassetts, EPL, Logit d2d) that support interoperability and harmonizing logistics processes in group of enterprises and as a result allows joint organization of haulages. Furthermore the e-platforms is to support standardized data exchange process thereby enabling cooperation between users and transport service provider for all modes.

4 Coordination Mechanism and Trust Creation

As mentioned before the participants of intermodal transport network should use web-based platform for exchange of the information. In order to gain information the Coordination Agent (CA) delegates collecting of predefined information scope about potential business partners to group of mobile agents. CA represents the particular interests of cooperating companies and it is responsible for cooperation, communication, negotiation. On the basis of collected information the CA makes decisions regarding follow-up actions. Coordination Agent is able to manage complex business processes. It cooperates with groups of agents and creates multi-agent system, as followed:

- Users Agent (UA) – it is responsible for issuing the demand and the definition of criteria for different cooperation scenario. It is able to inform the Verifying Agent about semantic inconsistency of found information.
- Logistics Providers Agent (LPA) – represents companies which offer particular transport services
- Informing Agent (IA) – it is responsible for sending information to CA about new attractive offers available, as well as for informing clients agents and transport providers agent about the possibility of starting cooperation with CA after fulfillment of defined criteria.
- Verifying Agent (VA) – it verifies ontologies and acts together with tool for semantic verification of information placed in the web by trading partners.

Agents representing particular companies cooperate together and negotiate cooperation conditions in order to reach common goal and at the same time to fulfill the outlines of delegating institutions. It should be mentioned that agents representing different, sometimes competing companies can build up temporary alliances in order to reduce the cost of logistics services and environmental impact.

The communication between agents and user is asynchronous, so the confirmation of message by receiver is not required. Agents and users are informed automatically about any new event e.g. new offer arrival. The access to the semantic web does not require application of any specialized IT systems. Information can be process in format that is readable and understandable for computer and any user. This solution gives SMEs a better possibility to capture information and exchange them within particular enterprises' network, as well as it helps to build up business relations [3].

Application of the agent-based system and semantic web allows the improvement of the coordination of processes in virtual supply chain. The coordination mechanism follow the steps presented below [modified from 2]:

1. Users of logistics services order their agents (UA) issuing demand for transport (or other additional) services and a choice of carrier for shipment of goods according to sustainability and economic optimization criteria. The users form virtual network in order to combine shipments to the same geographical region.
2. UA through hyperlinks finds ontology defining key words. Then it communicates with LPA representing potential service providers and collects their offers and passes them to CA for analysis.
3. In case when UA has doubts about the content of the offer, it could command Verifying Agent (VA) to verify ontology (Common Transportation Framework) and to contact its issuer.
4. Before offer choice CA questions providers agent LPA whether the offer is still valid. If not, then the whole procedure needs to be repeated. If the offer is still valid, then the appropriate resources are booked. CA can book multiple service providers, especially in case when individual carrier is not able to provide intermodal transport service.
5. After the offer is chosen CA informs agents representing business partners UA and TPA and starts negotiations regarding number of routes, delivery dates, prices etc.
6. If the negotiations are successful, then the cooperation starts and agents confirm the service purchase on behalf of companies they represent.
7. When the transaction is finished, CA stores the scenario in its database in order to simplify the negotiation process in the future.

The problem of trust creation among virtual supply chain partners is crucial. The main role should play the coordinator of transport and logistics processes (see reference model). He is responsible not only for coordination of logistics processes but also should analyze all the evaluations of services which are made by users. The application of security mechanisms hinders service providers from manipulating their ratings respectively [1]. Prerequisite for the correct functioning of the trust procedure is the availability of a Public-Key-Pair for all participants with certificates [1]. The protocol is presented in figure 3.

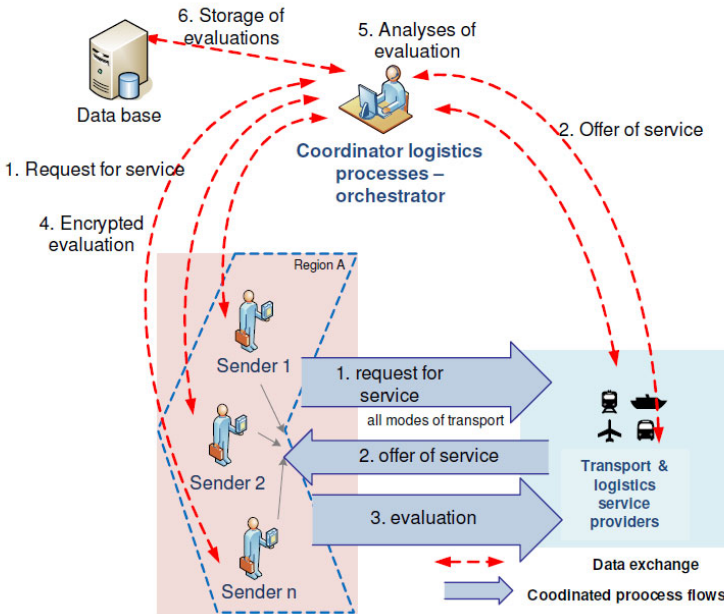


Fig. 3. Trust model

The protocol includes the following steps (see Fig.3):

1. The service user requests the execution of a service - depending on the size of order it might by send individually or coordinator of logistics processes might combine some smaller orders (virtual supply chain) and issue order.
2. The service provider executes the service and asks for the evaluation of his service.
3. The service user (or network of users) signs the encrypted evaluation in order to identify himself against both the provider and the coordinator. Then the service user sends his encrypted evaluation to the provider.
4. The encrypted evaluation and the key are sent to the service coordinator.
5. The coordinator is responsible for decryption and evaluation of service providers on request of both users and provider. The provider can update his evaluation summary sending all his periodic encrypted evaluations to the coordinator .

- The coordinator checks the validity of all signatures, decrypts the evaluations and compares all evaluations of the past, which are stored in the database with the new ones. The results are updated. The new evaluation summary is sent back to the provider and network of service users (virtual supply chain).

5 Validation of Coordination Mechanism in Pharmacy Sector

Validation of created reference model was carried out within **DiSCwise project**, founded by DG Enterprise, European Commission in 2010-2011. DiSCwise project aims to Develop, Demonstrate and Deploy a Reference Architecture for Interoperability in the Transport and Logistics Sector in an effort to achieve:

- Integration of small and medium sized transport service providers into efficient door-to-door supply chains at cost affordable to them.
- Enabling transport users to select environment-friendly alternatives.
- Making European supply chains more efficient by providing state-of-the-art tracking and management capabilities.

Figures 4-5 presents scope of cooperation between suppliers, wholesalers, logistics service providers in order to reduce transport costs.

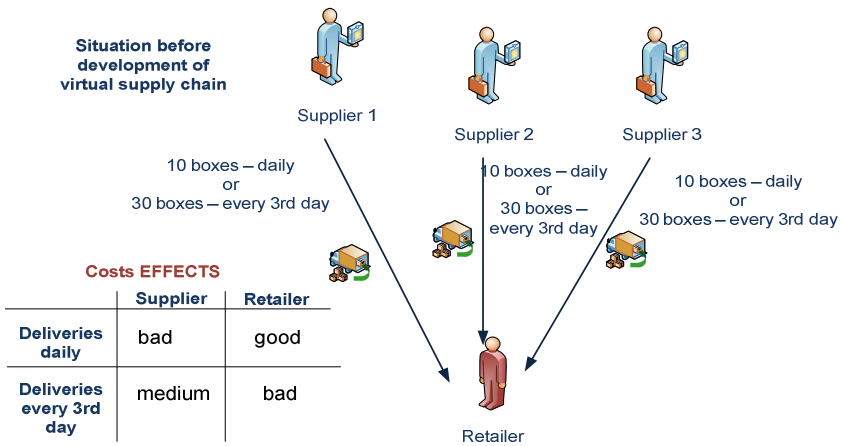


Fig. 4. Current transport business model analysed in DiSCwise demonstration

The DiSCwise project is funded by DG Enterprise and supports EU’s Freight Transport Logistics Action Plan. One of the demonstration in DiSCwise projects, have been carried out by the Institute of Logistics and Warehousing, Logit d2d and virtual pharmacy cluster in Poland.

First tests were carried out on a small sample representing group of four distributors in Poznan, Poland. Before implementation of new organisational solution, companies did not cooperate at any level, but had deliveries to the same clients or clients located very close to each other.

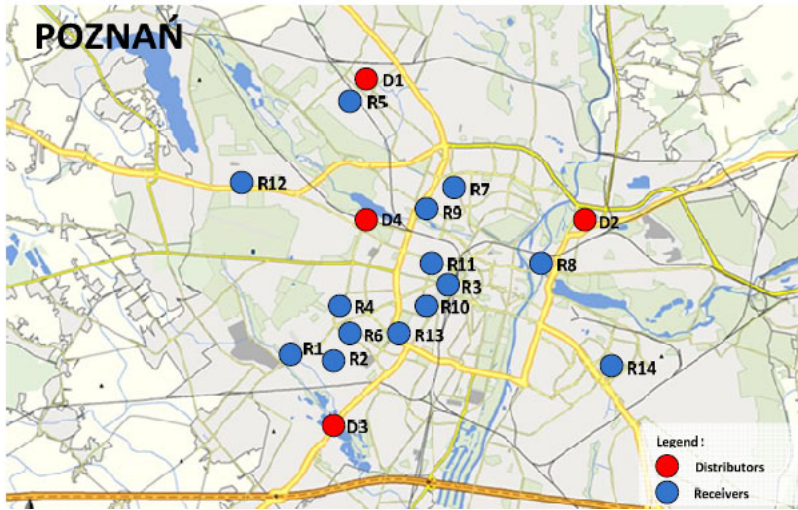


Fig. 5. Localization of distributors and delivery

In analysed case illustrating average day, each of the chemists ordered selected amount of products, which were delivered in special boxes (ca. 30 l), dedicated and standardized for pharmacy sector. Below implementation results are presented.

Table 1. Simulation results for two analysed options

Distributors	Lack of cooperation			Cooperation (new model implemented)			
	Transport cost KKTn (1 tkm = 1,08 PLN)	Average transport cost of single box [PLN]	Number of used trucks [PCS]	Transport cost KKTn (1 tkm = 1,08 PLN)	Average transport cost of single box [PLN]	Percentage costs savings [%]	Number of used trucks [PCS]
D 1	19,7	6,57	1	15,1	5,02	23,6	1
D 2	34,3	6,87	1	26,6	5,32	22,6	
D 3	26,3	5,26	1	18,5	3,70	29,5	
D 4	49,7	8,29	1	40,4	6,73	18,7	
Total	130,1	6,85	4	100,6	5,29	22,7	1

The table above shows precisely the profits from the implemented solution. What is also worth paying attention to is that one truck is used instead of four ones which decreases road traffic congestion. Therefore, the solution eliminates disadvantages of the traditional method of transport process organization.

6 Conclusions

Almost unlimited access to the Internet makes possible cooperation in the area of logistics process not only between big company but also SMEs. The lack of consistency of

business process performed by particular entities and the variety IT systems used by companies, cause problems with automatic partners networking. The presented by authors coordination mechanism provides framework for planning and coordination of intermodal transport in micro (company) and macro scale (region). Proposed solution enables efficient creation and automatic collection of data about companies and their resources published in the Internet. Agents representing particular companies coordinate and establish cooperation's conditions, in order to reach common goal of cost effectiveness as well as the sustainable development goals.

References

- [1] Brehm, N., Golinska, P.: Engineering Web Service Markets for Federated Business Applications. In: Omatu, S., Rocha, M.P., Bravo, J., Fernández, F., Corchado, E., Bustillo, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5518, pp. 366–373. Springer, Heidelberg (2009)
- [2] Kawa, A., Pawlewski, P., Golinska, P., Hajdul, M.: Cooperative purchasing of logistics services among manufacturing companies based on semantic web and multi-agents system. In: Demazeau, et al. (eds.) Trends in Practical Applications of Agents and Multiagents Systems. AISC (Advances in Intelligent and Soft Computing), vol. 71, pp. 249–256. Springer, Heidelberg (2010)
- [3] Kisperska-Moroń, D. (red.): Czynniki rozwoju wirtualnych łańcuchów dostaw, Wydawnictwo Akademii Ekonomicznej w Katowicach, Katowice (2009)
- [4] Pedersen, T.J., Paganelli, P., Knoors, F.: One Common Framework for Information and Communication Systems in Transport and Logistics, DiSCwise project deliverable, Brussels (2010)
- [5] Ulieru, M., Brennan, R.W., Walker, S.S.: The Holonic enterprise: A model for Internet-enabled global manufacturing supply chain and workflow management. *Integrated Manufacturing Systems* 13(8) (2002)
- [6] <http://www.gs1.org>

Competence-Based Architecture for Knowledge Logistics in Project-Oriented Organization

Przemysław Różewski and Bartłomiej Małachowski

West Pomeranian University of Technology in Szczecin,
Faculty of Computer Science and Information Systems,
ul. Żołnierska 49, 71-210 Szczecin, Poland
{prozewski, bmalachowski}@wi.zut.edu.pl

Abstract. Many modern knowledge-based organizations (i.a. IT companies) apply the project-oriented model to manage all their business activities. Projects run within knowledge-based organizations consisting primarily of intellectual tasks require adequate knowledge from the organization staff. The knowledge logistics process assumes that the knowledge has to be delivered in the right context to the right task/project/person, in the right time for the right purpose. In authors opinion the competence management can play important role in the knowledge logistics process. The paper presents the competence-based architecture (CBA) adapted to knowledge logistics process in the project-oriented organization.

Keywords: knowledge logistics, competence management, knowledge-based organizations.

1 Introduction

Most of the modern companies, especially in the hi-tech sector, can be characterized as knowledge-based organizations. In these organizations knowledge workers deal with knowledge according to workflow rules, performing different intellectual activities for tacit and explicit knowledge processing: capture/extract, analyze/organize, find, create/synthesize, distribute/share [5]. The knowledge workers are interconnected within the knowledge network. In the knowledge network the knowledge workers (nodes) actively process knowledge and communication and social links (edges) represent channels for knowledge relocation. The one of the most important dynamic aspects of knowledge management in the knowledge network framework is the knowledge logistics [1].

The knowledge logistics is an important research issue in the knowledge management [12,17] that applies methods of the “classic” logistics to model and control the flow of outcomes of intellectual activities. The knowledge logistics process deals with effective knowledge delivery process. Under the effective knowledge logistics process all available knowledge should be properly represented, correlated, and accessed in knowledge network [7]. The knowledge logistics process covers acquisition, integration, and transfer activities and assumes that the knowledge has to be delivered in the right context to the right task/project/person, in the right time for the right purpose [1,17].

So far two mature approaches to the knowledge logistics process can be found in the literature. The main difference are the knowledge model and the working environment:

1. Knowledge model: ontology. Environment: multi-agents system.

The knowledge logistics process based on the idea of adaptive hypermedia systems, where the knowledge logistics process provides personalized information that depends on the users' needs and requirements [12]. In this approach the knowledge logistics process dealing with knowledge fusion from distributed heterogeneous knowledge sources [12]. Moreover the knowledge logistics process provides a set of activities for knowledge search, acquisition and integration from distributed sources located in information grid environment [13].

2. Knowledge model: knowledge grid [18]. Environment; knowledge flow network.

The knowledge logistics process takes place in a knowledge service chain or network where nodes are team members, software agents or knowledge portals that provide services, and the links are flows of knowledge between nodes [17,18]. The logistics knowledge process organizes the knowledge flow network in the efficient and effective way that the network is free of unnecessary flows. The relation's interpretation between workflows, team members, roles and knowledge flows is a key issue in the knowledge logistics process [17]. The proposed framework is focused on the knowledge network described by associated workflow and used multilayer Knowledge Grid approach for implementation.

In the above presented approaches the quantitative analysis of the knowledge logistics processes as well as use of quantitative measures of knowledge are very limited. In both of these approaches the abstract notion of knowledge can be translated in operational knowledge objects [15]. Where the knowledge objects reflect only the explicit aspects of organization knowledge. The proposed approach is aimed to eliminate this limitation.

In authors opinion the competence management can play important role in the knowledge logistics process due to better adaptation of the knowledge flow to the knowledge node's profile. Many works have shown that the application of competence in companies allows to optimized task time, provide more adequate human resources management and more efficient knowledge transfer process [9, 10]. The concept of competence is the background for the competence management. According to ISO 9000:2005 the competence is "demonstrated ability to apply knowledge and skills". The authors use the (fuzzy) competence set theory in order to perform quantitative calculations about competence [16]. Moreover, the fuzzy competence set theory allows every knowledge worker, task or project to be described by the fuzzy competence set [4] for which the competence expansion cost can be calculated. With this feature, the approach provides analytical background for development of more accurate methods of knowledge logistics. In the competence-based approach to the knowledge logistics we operate with the competence set on the operational level. The competence nature allows to model the tacit knowledge as well as explicit knowledge in the form of the fuzzy competence sets.

The main goal of the paper is to show how the knowledge logistics process can be expanded by the methods of competence management. The competence-based

methodology for knowledge logistics process is presented. Moreover, the paper presents the competence-based architecture (CBA) adapted to knowledge logistics process in the project-oriented organization, which is based on the fuzzy competence set theory applied in the knowledge network environment. The architecture has been partially implemented in the Competence Object Library. In order to demonstrate the approach the competence editor tool was developed. The CBA framework integrated author's earlier works in the field of the competence and knowledge management [4,8,9].

2 Project-Oriented Organization

Many modern organizations, whose core business activities are creation and processing of knowledge, use project-oriented management (sometimes also referred as 'management by projects') as their primary management model. This follows from the nature of these activities, which hardly ever can be defined as a continuous processes (e.g. manufacturing) containing static action sequences and a permanently allocated resources. Organizations of this type lack classic production resources and their main potential are human resources with their intellectual capital. This situation creates specific requirements for managements methods, among which the methods for human resource management seem to have the strategic importance.

Review of project management literature shows a big gap in publications in the field of management of project-oriented organizations [3]. However, there are plenty of articles, books and even sophisticated methodologies (e.g. PRINCE2) but they focus mainly on single project management throughout all its phases. At the same time there are not many methods for integrated management and coordination of simultaneously conducted projects. Moreover, if we narrow down the scope of literature analysis only to knowledge-based organizations the lack of suitable methods for project-oriented management will be even more visible.

In order to describe the project-oriented organization we can use definition found in [2], according to which the project-oriented organization has the following characteristic: (i) defines "management by projects" as its organizational strategy; (ii) applies temporary organizations for the performance of complex processes; (iii) manages a project portfolio of different internal and external project types; (iv) has specific permanent organizations like a project portfolio group, or a PM office to provide integrative functions; (v) views the organization as being project-oriented. The concept of knowledge-based and project-oriented organization presented in the list above can be extended with the following assumptions: (vi) Tasks and projects implemented within the organization have for the most part an intellectual nature. This follows from the fact that the majority of tasks are untypical or innovative and require employing intellectual effort in order to successfully implement them. (vii) Human resources are nonhierarchical and comprise of independent workers, whose the main characteristics is their knowledge. Because of the internal dynamism of the project-oriented and knowledge-based organization, there is no static organizational structure for establishing a permanent position and responsibilities. In the case of such an organization we are dealing with temporary organizational structures conditioned by the currently implemented projects.

The internal structural dynamism of project oriented organization requires special methods for human resource assignment [3]. In this case human resource assignment and employment is the continuous process, which has to deliver personnel for every project run within the organization. Because of the intellectual nature of the majority of project tasks human resource assignments should be performed basing on the competence criterion. In this case the competence of the knowledge worker can be understood as the degree of matching person's knowledge to a task, which is mainly intellectual and require certain amount of knowledge in order to be successfully accomplished. Implementation of such a criterion requires exact quantitative measures of competence and formal model for their processing.

3 Competence Management in the Project-Oriented Organization

The competence management process is based on the knowledge management process due to their similar nature. Both processes use the same strategy (Tab. 1) with regard to different object. To fully understand the competence management the term of competence set has to be defined. The competence set has its roots in the habitual domain approach like the knowledge object has its root in the ontology. The habitual domain reflects the person's unique set of behavioral patterns resulting from his or her ways of thinking, judging, responding, and handling problems, which gradually stabilized within a certain boundary over a period of time [16].

Table 1. The knowledge and competence management similarity

	Knowledge management	Competence management
Objects	Knowledge object [15].	Competence set [4,16].
Codification strategy (person-to-document approach) Based on the externalization process (tacit to explicit) [6].	Encodes and stores knowledge in online databases and various repositories where it can be easily used by any knowledge worker [18].	Encodes and stores competence in online databases and various repositories where it can be easily used by any knowledge worker
Personalization strategy (person-to-person approach) Based on the socialization process (tacit to tacit) [6].	Creates, uses and shares knowledge peer-to-peer, supported by appropriate communication facilities [18]	Creates, uses and shares knowledge peer-to-peer, based on the competence profile, supported by appropriate communication facilities

The most important scientific result related to competence concept is a common understanding of complex competence nature. One of the approaches, from TEN-Competence project [14], describes competence structure as an aggregate of: proficiency level, context, description. In addition competence structure has been extended by the relation strength based on the fuzzy competence set theory [4].

4 Knowledge Logistics Process Based on the Competence Management

4.1 The Knowledge Source Selection Problem

In project-oriented organisations knowledge is considered as main resource requiring efficient methods of management, as well as main estimate of organisation value. According to [18] two basic strategies for knowledge management are *codification* and *personalisation*. The codification (way of knowledge description and storing) strategy is supported by information technologies such as databases, data warehouses, group support systems, Web browsers, search engines, knowledge bases, workflow management systems, and knowledge interchange formats [18]. The personalisation strategy is reached by associating knowledge not with forms of knowledge storage (databases, documents, publications etc.) but only with competent human resources.

The aim of knowledge based and project oriented organisation is to apply its knowledge to solve intellectual and innovative problems and tasks. The market value of company knowledge can be only evaluated in the context of its relevance to solving problems suitable for commercialisation.

The competence seen as a capacity for practical application of knowledge can be used as a measure of knowledge and meet requirements of codification strategy for knowledge management. Thus, the knowledge of organisation can be evaluated by assessing competence (codification) of its human resources (personalisation).

In the context of knowledge management one can model organisation as a network of individual knowledge workers. Within such a networked structure exists knowledge flows caused by local demands for competence. These demands result from requirements of tasks assigned to knowledge workers within current project run by the company.

Managing knowledge flows within the organization requires methods of knowledge logistics allowing to find and provide the right knowledge to the right person at the right time. According to [18] the knowledge flow has three main attributes: direction, content and carrier. Direction of knowledge flows is set by knowledge potentials described by a specific method of codification. The content is described by knowledge demanded in a node, whereas carrier can be any method of personal communication.

These assumptions provide a solid background for applying competence management methods as tools for knowledge flows management. The set of competences demanded by a person can be modelled as the required competence set for accomplishing the assigned task. The selection of a person which can be the source of demanded competence can be achieved through comparison of all network nodes (knowledge workers) to required competences for the task and, thus finding the node from which the knowledge flow can be initiated. This can be formalized as follows:

$P = \{p_i\}$, $i = 1, \dots, I$ - set of knowledge workers within an organisation

T - problem or task to be accomplished.

$D = \{d_k\}$, $k = 1, \dots, K$ - set of competences related to domain containing the problem T .

$[r_{xy}]^{K \times K}$ - relation matrix of competences in the set D .

$C(p_i) = C_i = \{(d_n, \alpha_i(d_n)) : d_n \in D\}$ - fuzzy competence set of i -th person, where:

$\alpha_i : d_n \rightarrow [0,1]$ - membership function of d_n in competence set C_i describing the strength of competence d_i of person p_k .

$R_T = \{(d_m, \beta(d_m)) : d_m \in D\}$ - fuzzy set of competences required to accomplish the task T , where:

$\beta : d_m \rightarrow [0,1]$ - membership function of d_m in competence set R describing the required strength of competence d_m .

$c(R_T, C_i) = c_i^T$ - scalar measure of person's p_i lack of competence to accomplish the problem T , where $c : R_T \setminus C_i \rightarrow R^+$.

Thus, the choice of a person, which can be used as a source of competence within the network of intellectual workers can be done on the basis of the measure of personal lack of competence to accomplish a given task. The person with the minimum value of the 'lack of competence' measure should be selected as trainer or tutor that can assist the person responsible for the task in carrying out his or her job. In other words, the person with the lowest 'lack of competence' across the network should be selected to initiate the knowledge flow oriented towards the person demanding new competences. However, in real life conditions the selection of a person, which can be knowledge source for his/her colleagues may be limited by several constraints. For example a worker can be temporally unavailable due to other duties or assignments. Moreover, the competent worker can be inaccessible for some other workers because of the organizational structures and lack of communication channels that can be used.

Taking into account the above limitations, competence based knowledge source selection criterion can be formulated as follows:

$$(c(R_T, C(p_x)) - c(R_T, C(p_i))) \cdot a_i \cdot s_{xi} \rightarrow \max. \tag{1}$$

where: p_x – person requiring new knowledge;

p_i – person being potential knowledge source, $i = 1, \dots, I \wedge i \neq x$

a_i – availability of person p_i , $a_i = \begin{cases} 1 & p_i \text{ is available} \\ 0 & p_i \text{ is not available} \end{cases}$

s_{xi} – accessibility of person p_i to person p_x , $s_{xi} = \begin{cases} 1 & p_i \text{ has access to } p_x \\ 0 & \text{otherwise} \end{cases}$

4.2 Multiple Criteria Decision Support Model for Knowledge Source Selection

The model of selecting knowledge source for a person requiring new competence described in the Section 4.1 assumes, that the decision about the assignment is made basing on only one criterion. This criterion illustrates the potential of a person to share some knowledge from a certain domain. In real life situations the nature of decision making is more complex and comprises several criteria having different scales, units, quantitative or qualitative measures [11]. For the problem of knowledge source selection we propose knowledge model comprising three criteria:

1. **Domain competences criterion** – indicating the node with the highest level of competence according to current competence demands. Formalisation in Section 4.1 modelled this criterion as quantitative function of competences possessed by knowledge workers.
2. **Knowledge sharing skills criterion** – determining the ability of people to transfer knowledge to others. This capacity, requiring specific skills can be assessed by HR specialist using several methods providing different measures (qualitative or quantitative).
3. **Knowledge flow costs** – competence transfers from one knowledge worker to another can generate costs. Many different organisation-specific cost models can be applied here.

The analytic hierarchy process (AHP) is an intuitively easy method for formulating and analyzing decisions. In series of pair wise comparisons it allows to obtain criteria weights and preference measures subjective to decision maker opinion. One of the most important quality of this method is, that it provides framework for integration of both tangible and intangible criteria based both on the judgment of knowledgeable and expert people and on existing measurements and statistics needed to make a decision [11].

5 Competence-Based Architecture for Knowledge Logistics

The Competence-based Architecture (CBA) is developed to support the knowledge logistics process. The presented CBA uses the fuzzy competence set theory and several information system modules. The competence set is an equivalent of knowledge objects in “classic” approach to the knowledge logistics. On the fig 1 the example competence management system based on the CBA is presented.

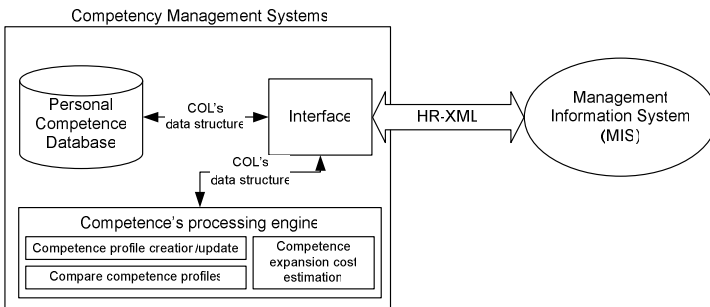


Fig. 1. The architecture of the competence management system

The CBA framework comprises several interoperating modules:

1. The competence processing engine includes a set of methods and algorithms and related structures for quantity competence processing. The engine provides method for the competence profile creation/update, comparison, and competence expansion

cost estimation algorithm [4]. The details of competence processing engine have been discussed in the authors' previous publications [4,9].

2. The personal competence database, which is responsible for competence profile storing for every knowledge workers, tasks and projects. The more information about the personal competence database can be found in [8].
3. The competence object library [9], which was developed to support competence modeling, development and processing.

The main CBA feature is the ability to quantify system estimations. The CBA provides quantitative characteristics about competences possessed by the organization, and competence required for tasks or projects. Moreover, the CBA supplies tools for quantitative estimation of competence expansion cost for knowledge workers and project groups.

In the next chapters already implemented elements of the CBS will be discussed in details. The competence object library (COL) is main interface mechanism inside the system. What is more, COL is used in personal competence database and personal competence profile development. The competence profile editor is required for communication outside the competence management system.

6 The Competence Object Library

The reusable Competence Object Library (COL) integrates objective data structures complying with TENCompetence Domain Model (TCDM) with method for fuzzy competence sets modeling and method of competence expansion cost analysis proposed in [4,9].

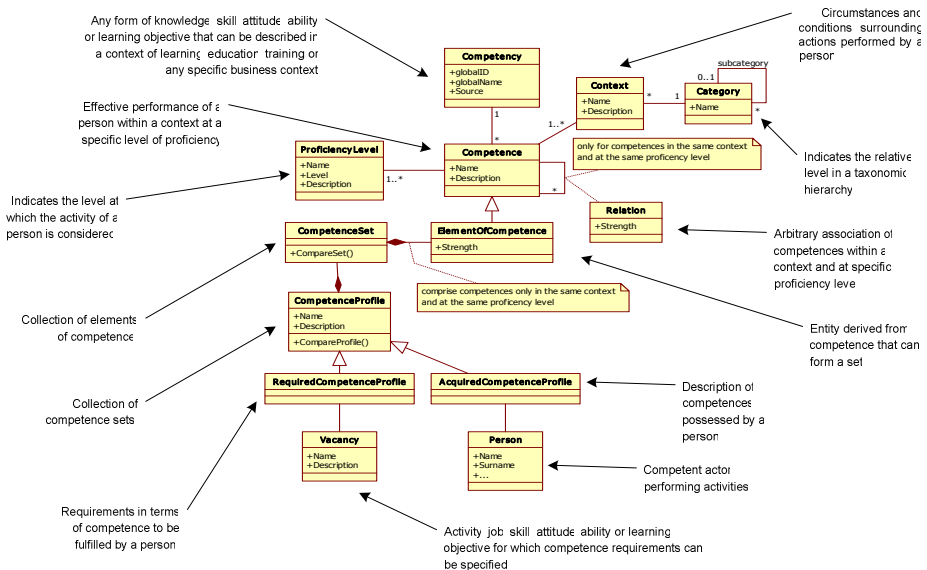


Fig. 2. UML Class Diagram for Competence Object Library

The proposed COL allows rapid development of competence management systems for different purposes and supports communication interface inside the competence management system. The COL structure is presented on the fig. 2 and in paper [9]. In COL elements of competence description have been connected with method for competence processing. The main object is a competence related to proficiency level and context. The competence is a part of competency, which can be treated as the habitual domain. The element of competence stores the competences with related strength attributes in the form of competence set. The competence set supplies the *CompetenceSet.CompareSet()* and *Competence-Profile.CompareProfile()* methods for competence expansion cost computing algorithm [4].

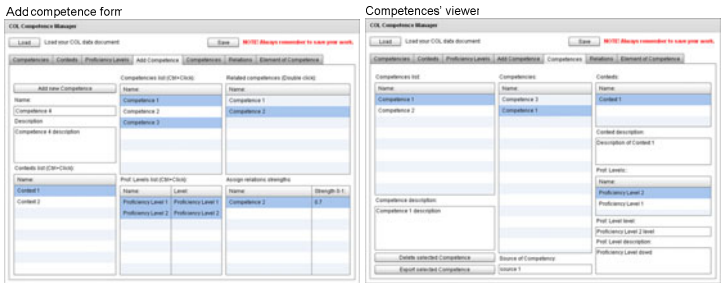


Fig. 3. Competence editor

The competence profile development is a base procedure for competence logistics process. For this purpose the competence editor was developed (fig 3). The editor allows editing and creating competence profiles according to COL and save competence profile in the personal competence database.

7 Conclusion

The paper presents the competence-based architecture (CBA) adapted to knowledge logistics in the project-oriented organization, basing on the fuzzy competence set theory. So far the competence processing engine has been developed. What is more the COL and competence editor has been implemented.

The CBA is adapted to the requirements of modern information system. Especially the COL allows rapid development of applications for competence management and analysis as a part of knowledge/competence management system. The discussed method deals with the knowledge logistics issue on the competence level. The competence-based approach to knowledge logistics is different form approaches developed by [17,18] or [12,13] because the new method is focused on the computational aspect of knowledge logistics. The CBA can be applied in different types of knowledge-based organizations.

Future work includes better correlation of knowledge logistics and knowledge network. The idea is to predict future problems in the filed of knowledge logistics basing on the workflow rules and information about product. Next, we want to apply CBA approach in other types of organizations (eg. virtual organization).

References

1. Alvarado, M., Sheremetov, L., Bañares-Alcántara, R., Cantú-Ortiz, F.: Current challenges and trends in intelligent computing and knowledge management in industry. *Knowledge and Information Systems* 12(2), 117–127 (2007)
2. Gareis, R., Huemann, M.: Project Management Competences in the Project-oriented Organisation. In: Turner, J.R., Simister, S.J. (eds.) *The Gower Handbook of Project Management*, pp. 709–721. Gower, Aldershot (2000)
3. Huemann, M., Keegan, A., Turner, J.R.: Human resource management in the project-oriented company: A review. *International Journal of Project Management* 25(3), 315–323 (2007)
4. Kusztna, E., Zaikin, O., Rózewski, P., Małachowski, B.: Cost estimation algorithm and decision-making model for curriculum modification in educational organization. *European Journal of Operational Research* 197(2), 752–763 (2009)
5. Mack, R., Ravin, Y., Byrd, R.J.: Knowledge portals and the emerging digital knowledge workplace. *IBM Systems Journal* 40(4), 925–955 (2001)
6. Marwick, A.D.: Knowledge management technology. *IBM Systems Journal* 40(4), 814–830 (2001)
7. Mulvenna, M.D., Zambonelli, F., Curran, K., Nugent, C.D.: Knowledge networks. In: Stavrakakis, I., Smirnov, M. (eds.) *WAC 2005. LNCS*, vol. 3854, pp. 99–114. Springer, Heidelberg (2006)
8. Rózewski, P., Małachowski, B.: System For Creative Distance Learning Environment Development Based On Competence Management. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) *KES 2010. LNCS (LNAI)*, vol. 6279, pp. 180–189. Springer, Heidelberg (2010)
9. Rózewski, P., Małachowski, B.: Competence Management In Knowledge-Based Organisation: Case Study Based On Higher Education Organisation. In: Karagiannis, D., Jin, Z. (eds.) *KSEM 2009. LNCS (LNAI)*, vol. 5914, pp. 358–369. Springer, Heidelberg (2009)
10. Sampson, D., Fytros, D.: Competence Models in Technology-Enhanced Competence-Based Learning. In: Adelsberger, H.H., Kinshuk, P.J.M., Sampson, D. (eds.) *Handbook on Information Technologies for Education and Training*, 2nd edn. pp. 155–177. Springer-Verlag, Heidelberg (2008)
11. Saaty, T.L.: The seven pillars of the analytic hierarchy process. In: Köksa-lan, M., Zions, S. (red.) *Multiple Criteria Decision Making in the New Millennium*. Springer, Heidelberg (2001)
12. Smirnov, V.A., Pashkin, M., Chilov, N., Levashova, T.: Knowledge logistics in information grid environment. *Future Generation Computer Systems* 20(1), 61–79 (2004)
13. Smirnov, A.V., Pashkin, M., Chilov, N., Levashova, T.: Multi-agent Knowledge Logistics System "KSNet": Implementation and Case Study for Coalition Operations. In: Mafík, V., Müller, J.P., Pěchouček, M. (eds.) *CEEMAS 2003. LNCS (LNAI)*, vol. 2691, p. 292. Springer, Heidelberg (2003)
14. TENCompetence - Building the European Network for Lifelong Competence Development, EU IST-TEL project (2005-2009), <http://www.tencompetence.org/>
15. Wijnhoven, F.: Operational knowledge management: identification of knowledge objects, operation methods, and goals and means for the support function. *The Journal of the Operational Research Society* 54, 194–203 (2003)
16. Yu, P.L., Zhang, D.: A foundation for competence set analysis. *Mathematical Social Sciences* 20, 251–299 (1990)
17. Zhuge, H.: Knowledge flow network planning and simulation. *Decision Support Systems* 42(2), 571–592 (2006)
18. Zhuge, H.: A knowledge flow model for peer-to-peer team knowledge sharing and management. *Expert Systems with Applications* 23(1), 23–30 (2002)

The Effects of Knowledge Management System Quality and Self-efficacy on Intention to Knowledge Sharing: The Mediate Role of Attitude

Shiuann-Shuoh Chen¹, Yu-Wei Chuang¹, and Pei-Yi Chen^{1,2}

¹ Department of Business Administration, National Central University, No. 300, Jung-da Rd., Jung-li City, Taoyuan 320, Republic of China Taiwan

kenchen@mgt.ncu.edu.tw, ywchuang@gmail.com

² Center for General Education, Hsin Sheng College of Medical Care and Management, No. 418, Gaoping Sec., Zhongfeng Rd., Longtan Township, Taoyuan 320, Republic of China Taiwan
peiyi01@ms47.hinet.net

Abstract. Knowledge sharing (KS) has been received increasing attention and view as a strategic tactic and knowledge management system (KMS) can play a value-added role in these activities. However, does KMS quality can enhance the knowledge sharing performance without considering other factors? For this reason, we not only take KMS quality as the given antecedent, but also take KMS self-efficacy and knowledge self-efficacy as precedent. Furthermore, we verify the attitude toward KS to understand its mediate effect in the model. Knowledge sharing is an emergent trend in real business practices. This study is catering to this trend and attempts to build a model capturing the dynamic of team collaboration and its consequences on intention to knowledge sharing. The results from this study expect to shed light on future KM research and offer practical guidelines for practitioners.

Keywords: knowledge management quality, self-efficacy, intention to knowledge sharing, attitude toward knowledge sharing, structural equation modeling.

1 Introduction

The current century is the era for digital economy. In the global tide, modern businesses face a high uncertainty and heavy variation of degree in the dynamic environment. How can firms achieve their competitive advantage? One possible way is via knowledge sharing (KS). The scholars of knowledge-based view (KBV) contented that knowledge is embedded in and carried by multiple entities, including individual members, groups, as well as organization wholes [1][17]. Knowledge is one kind of resources, having these characteristics of rare, valuable, nonsubstitutable, and difficult to imitate. If all the elements in organization can share and transfer their knowledge to each other, the competitive advantage will be enhanced.

Knowledge is usually tacit, complicated, and difficult to imitate in the organizations. The KBV insisted on that knowledge assets can yield long-term sustainable competitive advantage. For this reason, information technology, such as knowledge management systems (KMS), may play an important role in effectuating the KBV of

the firm [2]. However, does KMS itself can reinforce the knowledge sharing behavior in the organization without considering other factors? Is KMS the key element for knowledge sharing in practice?

Antecedent studies [1][24][25] have never thought of combining KMS factors and individual factors when they were studying the intention to KS; however, the KMS quality, self-efficacy, and knowledge self-efficacy factors will affect the intention to KS through other mediate variables in practice. Therefore, we can't thoroughly describe the actions in practice if we don't take individual factors into consideration.

Given the importance of organization knowledge, the objective of this paper is to synthesize the relevant and important knowledge management (KM) literature to contribute to and shape our understanding of KM and knowledge management systems (KMS) in organizations. In this article, we focus on how KMS quality, KMS self-efficacy, knowledge self-efficacy influence intention to knowledge sharing through attitude toward knowledge sharing.

The paper is organized as follows: the next section presents a review of theory and hypotheses. This section provides a comprehensive summary of the related KM literature used in this article. The following section presents the methodology of this research, including respondents and procedure and measures. The data analysis and result will present in section 4, comprising the analysis of the measurement model and test of the structural model. The final section provides a summary and presents the discussion and conclusion of our work.

2 Theory and Hypotheses

2.1 Knowledge Management System Quality and Attitude toward Knowledge Sharing

The knowledge engineering methodologies were usually adapted in KMS development and knowledge engineering projects to improve the knowledge-related processes and tasks [1][24] and accomplish the plans on time [25]. For the purpose of modeling knowledge, the most common method is CommonKADs, which is widespread accepted to be the standard in knowledge engineering projects [27]. The engineers could use the technique to conduct detailed analysis of knowledge processes and tasks, indicating that the design of KMS will be efficient and effective [1][27].

Following the definition of Kulkarni et al., [21], KMS quality include the accessibility, ease of use for retrieval and input, flexibility to meet the needs, search capability, documentation, and so on. A KMS, an IT-based system, is to support and enhance the organizational process of knowledge search, acquisition, repository, retrieval, and employment [2][19][28]. In organization, KMS is playing the supporting role for KM application [19][28], which could use the business intelligence technique, such as OLAP, data warehouse, and data mining to dig and search useful knowledge[28]. KMS quality in this research is focusing on how well the KMS assist and strengthen KM activities, which engage in capturing diverse scenarios of the quality of KMS. A well-built KMS can help individual members to search correct results to meet their needs. Under the situation, the quality of knowledge content and perceived usefulness

of knowledge sharing is the main consideration of KMS [21]. Thus, we believe that KMS quality has a positive effect on attitudes toward knowledge sharing.

H1: The greater the knowledge management system quality is, the greater the attitude toward knowledge sharing will be.

2.2 Knowledge Management System Self-efficacy and Attitude as Well as Intention toward Knowledge Sharing

Self-efficacy refers to “people’s judgments of their capabilities to organize and execute courses of action required to attain designated types of performance”, which clearly manifest that self-efficacy does not assess the actual skills that people hold, but evaluate the belief that people can accomplish [6]. Self-efficacy will affect people’s incentive, doggedness, endeavor to overcome difficulties, as well as perform successfully [16]. Individuals with stronger efficacy beliefs will drive them to make much effort and to incline more persistent [18].

KMS self-efficacy, then, involved the judgment of people’s ability to use KMS, which incorporates assessments of the capability to apply those skills to extended tasks [11]. Furthermore, Marakas et al., [23] indicated that computer self-efficacy not only influences people’s perceptions of their ability to accomplish a computer-related task, but also affects people’s attitude and intention toward future use of computers. Similarly, Deng et al, [14] suggested that computer self-efficacy influences an individual’s persistence, motivation, and creativity in computer use. In summary, we test the following hypotheses:

H2-1: The greater the KMS self-efficacy is, the greater the attitude toward knowledge sharing will be.

H2-2: The greater the KMS self-efficacy is, the greater the intention to knowledge sharing will be.

2.3 Knowledge Self-efficacy and the Attitude as Well as Intention toward Knowledge Sharing

Knowledge self-efficacy is individual’s confidence in his or her capability to provide valuable knowledge to other organization members [9]. People’s attitude and behavior are influenced by the factors from self-produced and the stimulation from external environment [8].

As people share useful knowledge to the organization, they will obtain confidence and believe that they can bring the benefits to the organization, which will increase their knowledge self-efficacy [12]. This belief can play a self-motivational driver for members to contribute knowledge to the community [7]. The person with higher knowledge self-efficacy, they will have much confidence to solve job-related problems [13] and to improve work efficiency [4]. Contrary, if people feel that they lack knowledge to contribute to the organization, they may not incline to contribute knowledge to the organization. Since they thought that their knowledge contribution cannot make a positive effect for the organization [18]. We therefore propose:

H3-1: The greater the knowledge self-efficacy is, the greater the attitude toward knowledge sharing will be.

H3-2: The greater the knowledge self-efficacy is, the greater the intention to knowledge sharing will be.

2.4 The Attitude toward Knowledge Sharing and the Intention to Knowledge Sharing

Attitude toward knowledge sharing is defined as the degree of one's positive feelings about sharing one's knowledge [8]. Ajzen and Fishbein [2] claimed that the intention to attend to a behavior is resulted from an individual's attitude toward that behavior. Following the logic, Chow & Chan [10] postulated that individuals' attitude toward a behavior can be taken as a predictor of intention to engage in that behavior. The intention to knowledge sharing is caused by a person's attitude toward knowledge sharing.

Also, Vijayasathy [29] indicated that individuals' beliefs about the outcomes will affect their attitude towards the behavior, and in turn influence their intention to perform that behavior. There are abundant theoretical and empirical evidence for the relationship between attitude and intention [4]. Thus, we propose hypothesis 4:

H4: The more favorable the attitude toward knowledge sharing is, the greater the intention to knowledge sharing will be.

3 Research Design and Methodology

3.1 Respondents and Procedure

In this survey, each respondent were asked to provide two different NPD projects, one of them has to be the superior performance project and another is inferior one, and then access their relationships respectively. Therefore, each questionnaire would consist of two samples in the survey. To assure the validity of respond data, we contacted each respondent by phone before sending the questionnaire to confirm whose position and task would have sufficient knowledge and ability to join this survey, and then we give a second call to reconfirm the willingness of the respondents to make sure that they would not answer it casually.

Two rounds of survey were conducted by distributing the survey instrument in the form of questionnaire to the production managers of 770 electrical manufacturing firms in Taiwan from June 1 to July 31, 2010. These firms were listed in the directories of the 2009 top 2000 firms in Chinese Credit (Taiwan's leading credit company). The exclusion of 13 invalid questionnaires resulted in a total of 156 complete and effective responses for data analysis (The first round yielded 102 effective responses, and the second round yielded an additional 54 responses.) The total response rate is 20.25%. A Chi-square analysis of the two rounds of the respondents showed no difference of all the firms used in the survey. This suggested no non-response bias in the returned questionnaires. Table 1 shows the demographic and characteristic profiles of participating firms.

Table 1. Profiles of participating firms

Demographic Profile	No. of firms	Percentage	Chi-square	d.f.	p value
<i>Tenure in present jobs</i>					
Under 5 years	98	31.4	5.559	4	0.235
6-10 years	33	10.6			
11-15 years	16	5.1			
16-20 years	4	1.3			
21-30 years	5	1.6			
<i>Department</i>					
R&D	77	24.7	6.657	5	0.247
Marketing	21	6.7			
Production	27	8.7			
Sales	15	4.8			
Purchasing	9	2.9			
Others	7	2.2			
<i>Total sales revenue (New Taiwan \$)</i>					
Under \$1 billion	60	19.2	7.665	6	0.264
\$1.1 billion to \$5 billion	30	9.6			
\$5 billion to \$10 billion	7	2.2			
\$10.1 billion to \$30 billion	14	4.5			
\$30.1 billion to \$50 billion	8	2.6			
\$50.1 billion to \$100 billion	8	2.6			
\$100.1 billion and above	29	9.3			

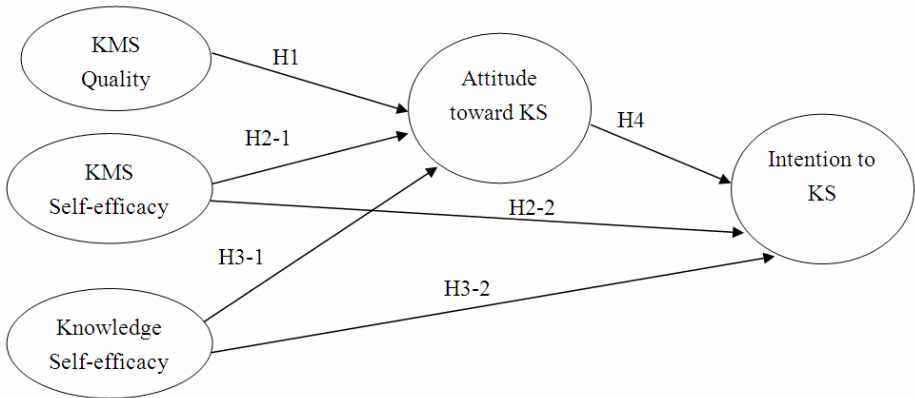


Fig. 1. The Conceptual Model

3.2 Measures

KMS Quality. Adapted from Kulkarni et al, [21], we identified three question items to measure the degree of KMS quality that induces the extent of KMS. **KMS Self-efficacy** According to Lin and Huang’s [22] work, we measure KMS self-efficacy by three items, such as capability to use KMS, understanding of KMS, and confidence in

using KMS. **Knowledge self-efficacy** Based on the previous study of Kankanhalli [20] we measure knowledge self-efficacy through the following items: confidence in ability to provide knowledge, expertise to provide knowledge, as well as ability to provide useful knowledge to other members. **Attitude toward KS** Measurement items were adapted from Bock et al, [8] to fit this study. We identify three items, enjoyable, valuable, and meaningful, to measure the construct. **Intention to KS** Adapted from Bock et al, [8], we take the following items to measure the construct: to share work reports and official documents, to share experience or know-how, and to share the knowledge from education or training. Drawing on relevant theories and evidence, the conceptual model of this research is proposed as shown Figure 1.

4 Data Analysis and Results

We used a two-step procedure for assessing the reliability and validity of the measures before their use in the full model.

4.1 Analysis of the Measurement Model

The measurement model was first evaluated and then respecified to generate the 'best fit' model. We used Amos 17.0 to evaluate the properties of the measures addressing latent constructs via confirmatory factor analysis (CFA). The construct unidimensionality, convergent, and discriminant validity of the candidate items and constructs used in a study were also assessed by employing latent variable constructs. The CFA results indicated that the measurement model fitted the data, $\chi^2/d.f.=2.01$, GFI=0.940, AGFI=0.910, NNFI=0.974, IFI=0.981, CFI=0.981 and RMSEA=0.057 as shown in Table 2. All the model-fit indices exceed the common acceptance levels, demonstrating that the measurement model exhibited a good fit with the data collected.

Table 2. Model fit results for measurement and structural models

Fit index	Measurement model	Structural model	Recommended cut-off value from literature
$\chi^2/d.f.$	161/80=2.01	161/81=1.98	$\leq 2^{**}$; $\leq 3^*$
GFI	0.940	0.939	$\geq 0.90^{**}$; $\geq 0.80^*$
AGFI	0.910	0.910	$\geq 0.90^{**}$; $\geq 0.80^*$
NNFI	0.974	0.975	$\geq 0.90^{**}$
IFI	0.981	0.981	$\geq 0.90^{**}$
CFI	0.981	0.981	$\geq 0.90^{**}$
RMSEA	0.057	0.057	$\leq 0.06^{**}$; $\leq 0.08^*$

Acceptability: ** acceptable, * marginal.

The convergent validity of the scales was verified by using normal three criteria: (1) that all indicator loadings should be significant and exceed 0.7, that the composite reliability (CR) should exceed 0.7, and that average variance extracted (AVE) by each construct should exceed the variance due to measurement error for that construct (i.e., should exceed 0.50). The CFA results were consistent with the relationships expected between the measured items and their respective constructs. All were significantly

related, via t-tests, to their respective constructs. From Table 3, the composite reliability of the constructs ranging from 0.87 to 0.95 all exceeded the benchmark of 0.7. AVE ranged from 0.70 to 0.86. Also, the internal consistency reliability to test unidimensionality was assessed by Cronbach's α . The resulting α values ranged from 0.85 to 0.95 (see Table 3), which were above the acceptable threshold 0.70. Because multivariate normality implies that the individual variables—and their combinations—are univariate normal, it is recommended that univariate normality of the variables be initially assessed. The result showed that none of the item parcels exhibits skewness or kurtosis values in excess of |2.0| (skewness range from -0.970 to -0.232 and kurtosis range from -0.952 to 0.696). Although there is no established cutoff indicating when the data deviate appreciably from multivariate normality, previous research has noted some significant problems arising when univariate skewness approaches |2.0| and kurtosis approaches |7.0|.

Table 3. Confirmatory factor analysis results of the measurement model

Construct	Internal Reliability		Convergent Validity		
	Item	Cronbach alpha	Factor Loading	CR	AVE
KMS Quality	KMSQ1	0.92	0.88	0.91	0.78
	KMSQ2		0.92		
	KMSQ3		0.86		
KMS Self-efficacy	KMSSE1	0.92	0.88	0.92	0.80
	KMSSE2		0.93		
	KMSSE3		0.88		
Knowledge Self-efficacy	KSE1	0.91	0.86	0.90	0.76
	KSE2		0.88		
	KSE3		0.89		
Attitude toward KS	AKS1	0.95	0.94	0.95	0.86
	AKS2		0.94		
	AKS3		0.91		
Intention to KS	IKS1	0.85	0.76	0.87	0.70
	IKS2		0.86		
	IKS3		0.89		

Notes: (1) CR= Composite Reliability; AVE= Average Variance Extracted. (2) See Appendix A for the description of each construct and item.

The discriminant validity of the scales was assessed by that the square root of the AVE from the construct was greater than the correlation shared between the construct and other constructs in the model. Table 4 shows the correlations among the constructs, with the square root of the AVE on the diagonal. All the diagonal values exceeded the correlations between any pair of constructs, indicating the measure has adequate discriminant validity.

In addition to discriminant validity assessment, we also checked for multicollinearity due to the relatively high correlations among some constructs. The variance inflation factor (VIF) is a useful statistic to assess such problems, with values below 3.3

Table 4. Discriminant Validity: Correlations and AVE

	Mean	St. dev.	AVE	KMSQ	KMSSE	AKS	AKS	IKS
KMSQ	4.22	1.64	0.78	0.88				
KMSSE	4.78	1.33	0.80	0.44	0.89			
KSE	4.81	1.19	0.76	0.30	0.61	0.87		
AKS	5.32	1.32	0.86	0.26	0.48	0.58	0.92	
IKS	5.04	1.15	0.70	0.28	0.58	0.67	0.58	0.83

Note: Diagonal elements are the square root of the average variance extracted (AVE). Off-diagonal elements are the correlations among constructs. For discriminant validity, AVE should be larger than squared correlation between any pair of constructs; hence diagonal elements should be larger than off-diagonal elements.

indicative of the absence of multicollinearity [15] (The VIF in this study range from 1.123 to 2.283). In summary, the measurement model demonstrated adequate and sufficient reliability, unidimensionality, convergent validity, and discriminant validity.

Finally, we conducted a test to assess if common method bias would be problematic in interpretation of the results. We used Harmon's single factor test, based on the assumption that if the risk of common method is substantial, a single latent factor will explain the majority of the covariance among the measures [26]. The single-factor model resulted in $\chi^2(90) = 1966$, compared with $\chi^2(80) = 161$ for the measurement model, indicating that common method bias was not a serious threat in this study.

4.2 Test of the Structural Model and Discussion

The causal structure of the hypothesized research model reflecting the assumed linear, causal relationships among the constructs was tested using the structural model. The Amos analysis was stated with the theoretical model, with three exogenous latent constructs (KMS quality, KMS SE and Knowledge SE) and two latent endogenous constructs (attitude toward KS and intention to KS). As shown in Table 2, all of the model-fit indices of the structural model exceeded their respective common acceptance levels: χ^2 to the degrees of freedom ratio of 1.98 ($\chi^2 = 161$; $df = 80$), GFI=0.939, AGFI=0.910, NNFI=0.975, IFI =0.981, CFI=0.981, and RMSEA=0.057, suggesting that the model fitted the data well. The resulting structural model, together with the path coefficients and their significance values is shown in Figure 2. One of six paths exhibited at least a p-value less than 0.05 level of significance.

As shown in past research, KMS quality was shown to be a strong predictor of attitude toward KS. However, the result indicated that there is non-significant relationship from KMS quality to attitude toward KS. Consequently, hypothesis H1 was not supported. With regard to H2-1 and H2-2, KMS SE had positively related to attitude toward to KS (path coefficient=0.168, $p<0.05$) and had positively related to intention to KS (path coefficient=0.197, $p<0.01$). Consequently, hypotheses H2-1 and H2-2 were empirically supported. In terms of H3-1 and H3-2, knowledge self-efficacy

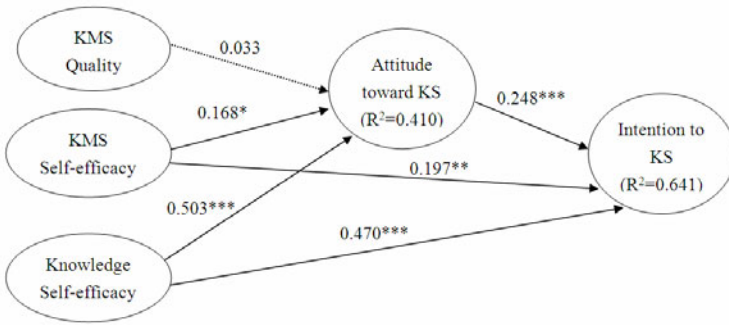


Fig. 2. Result of structural equation model (SEM) analysis

Note: *, **, and *** denote significance at p<0.05, p<0.01, and p<0.001 respectively.

had quite a strong positive direct effect on attitude toward KS (path coefficient=0.503, p<0.001) and intention on KS (path coefficient=0.470, p<0.001) Hence, hypothesis H3-1 and H3-2 were supported. Finally, the attitude toward KS had a stronger effects on intention to KS (path coefficient=0.248, p<0.001); therefore, hypothesis H4 was supported. The explanatory power of the research model is also shown in Figure 2.

5 Conclusion

From the research result, we can understand that KMS is an important mechanism for firms to facilitate knowledge sharing activities; however, firms should not focus on KMS itself, such as KMS quality, instead of concentrating on the self-efficacy of individual members. If employees have high KMS and knowledge self-efficacy, the proclivity of KS and the intention to KS will be lifted. Under the situation, KMS quality does not show significant effect on attitude toward KS. Also, we find that KMS and knowledge self-efficacy will directly influence intention to KS. Finally, this study chooses team members in the NPD projects as an example to construct and examine the structure model and theoretical model fit of KMS quality, KMS and knowledge self-efficacy, attitude toward KS, and intention to KS. Moreover, this study provides a complete and examined theoretical framework to add more detailed information in related studies.

Reference

1. Abdullah, M.S., Benest, I., Evans, A., Kimble, C.: Knowledge Modelling Techniques for Developing Knowledge Management Systems. In: 3rd European Conference on Knowledge Management, Dublin, Ireland (2002)
2. Alavi, M., Leidner, D.E.: Review: Knowledge management and knowledge management systems: conceptual foundations and research Issues. *MIS Quarterly* 25(1), 117–136 (2001)
3. Ajzen, I., Fishbein, M.: Understanding attitudes and predicting social behavior. Prentice Hall, Englewood Cliffs (1980)

4. Ba, S., Stallaert, J., Whinston, A.B.: Research commentary: introducing a third dimension in information systems design—the case for incentive alignment. *Information Systems Research* 12(3), 225–239 (2001)
5. Bajaj, A., Nidumolu, S.R.: A feedback model to understand information system usage. *Information & Management* 33, 213–224 (1998)
6. Bandura, A.: *Social foundations of thought and action: a social cognitive theory*. Prentice Hall, Englewood Cliffs (1986)
7. Bock, G.W., Kim, Y.G.: Breaking the myths of rewards: an exploratory study of attitudes about knowledge sharing. *Information Resource Management Journal* 15(2), 14–21 (2002)
8. Bock, G.W., Zmud, R.W., Lee, J.N.: Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate. *MIS Quarterly* 29(1), 87–111 (2005)
9. Chen, C.J., Hung, S.W.: To give or to receive? factors influencing members' knowledge sharing and community promotion in professional virtual communities. *Information & Management* 47, 226–236 (2010)
10. Chow, W.S., Chan, L.S.: Social network, social trust and shared goals in organizational knowledge sharing. *Information & Management* 45, 458–465 (2008)
11. Compeau, D.R., Higgins, C.A.: Computer self-efficacy: development of a measure and initial test. *MIS Quarterly* 19(2), 189–211 (1995)
12. Constant, D., Kiesler, S., Sproull, L.: What's mine is ours or is it? a study of attitudes about information sharing. *Information Systems Research* 5(4), 400–421 (1994)
13. Constant, D., Sproull, L., Kiesler, S.: The kindness of strangers: the usefulness of electronic weak ties for technical advice. *Organization Science* 7(2), 119–135 (1996)
14. Deng, X., Doll, W.J., Truong, D.: Computer self-efficacy in an ongoing use context. *Behavior and Information Technology* 23, 395–412 (2004)
15. Diamantopoulos, A., Siguaw, J.A.: Formative versus reflective indicators in organizational measure development: a comparison and empirical illustration. *British Journal of Management* 17(4), 263–282 (2006)
16. Gist, M.E.: Self-efficacy: implications for organizational behavior and human resource management. *Academy of Management Review* 12(3), 472–485 (1987)
17. Grant, R.M.: Toward a knowledge-based theory of the firm. *Strategic Management Journal* 17, 109–122 (1996)
18. Hasan, B.: Delineating the effects of general and system-specific computer self-efficacy beliefs on IS acceptance. *Information & Management* 43, 565–571 (2006)
19. Inmon, W.H., O'Neil, B., Fryman, L.: *Knowledge Management and Business Metadata*. In: *Business Metadata: Capturing Enterprise Knowledge*, ch. 15. Morgan Kaufmann, San Francisco (2007)
20. Kankanhalli, A., Tan, B.C.Y., Wei, K.K.: Contributing knowledge to electronic knowledge repositories: an empirical investigation. *MIS Quarterly* 29(1), 113–143 (2005)
21. Kulkarni, U.R., Ravidran, S., Freeze, R.: A knowledge management success model: theoretical development and empirical validation. *Journal of Management Information Systems* 23(3), 309–347 (2007)
22. Lin, T.C., Huang, C.C.: Understanding knowledge management system usage antecedents: an integration of social cognitive theory and task technology fit. *Information & Management* 45, 410–417 (2008)
23. Marakas, G.M., Yi, M.Y., Johnson, R.D.: The multilevel and multifaceted character of computer self-efficacy: toward clarification of the construct and an integrative framework for research. *Information Systems Research* 9(2), 126–163 (1998)

24. Papavassiliou, G., Mentzas, G., Abecker, A.: Integrating Knowledge Modelling in Business Process Management. In: European Conference on Information Systems, Gdańsk, Poland (2002)
25. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. In: 16th European Conference on Artificial Intelligence, Valencia, Spain (2004)
26. Podsakoff, P.M., MacKenzie, S.B., Lee, J.Y., Podsakoff, N.P.: Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of Applied Psychology* 88(5), 879–903 (2003)
27. Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R.D., Shadbolt, N., Van de Velde, W., Wielinga, B.: *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press, Cambridge (1999)
28. Teorey, T.J., Lightstone, S., Nadeau, T.: Business Intelligence. In: *Database Modeling and Design: Logical Design*, ch. 8, Morgan Kaufmann, San Francisco (2005)
29. Vijayarathy, L.R.: Predicting consumer intentions to use on-line shopping the case for an augmented technology acceptance model. *Information & Management* 41, 747–762 (2004)

Social Network Analysis as a Tool for Improving Enterprise Architecture

Przemysław Kazienko, Radosław Michalski, and Sebastian Palus

Institute of Informatics, Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{kazienko,radoslaw.michalski,sebastian.palus}@pwr.wroc.pl

Abstract. The paper provides the overview of essential analyses and methods, helpful for enterprise architecture improvement and based on social network approach. The ideas presented in this paper focus on social network, that is built with the use of real-life manufacturing company data. It has been shown that corporate social network analysis, as a decision support system, may be influential for managing a company. Several ideas, measurements, interpretations and evaluation methods are given and discussed, in particular centrality degree, social network extraction, process management.

1 Introduction

Nowadays, in the times of strong competition, business organizations constantly look for tools and techniques to beat market opponents and become leaders among other companies. This paper focuses on corporate social network analysis as a possible way to improve enterprise architecture leading to the above mentioned goals. However, the proposed methods are suitable for all kind of organizations with the stable organizational structure, not only the commercial ones.

Social network analysis is the well-established way of analyzing relations between persons in the social network. These methods focus on determining user position, extracting groups of users, describing and analyzing changes in such kind of networks [12]. Among many types of social networks, like social web sites, instant messaging systems, it has been found useful to build and analyze corporate social networks [1], [17], because results of such analysis may be found very helpful in improving corporate management.

This paper presents findings achieved by using corporate social network analysis in the mid-sized manufacturing company and possible ways of their interpretation as well as further usage of these results. The source data gathered for building the social network were unique and novel in the field of corporate social network analysis, because in other papers [17], only some assumptions about organizational structure were made due to lack of full information. This time, the achieved results have been compared to reliable information about company structure to depict that it is possible to improve organization management using proposed set of techniques as a decision

support tool. It has also been shown that only some metrics used in social network analysis are valuable and suitable in the process of the evaluation of a company.

Section 2 provides introduction to social networks in the enterprise describing possible ways of gathering and extraction of social network data. Section 3 relates to the idea of corporate network analysis. In Section 4, authors present their results from social networks analysis in the above mentioned company, while Section 5 focuses on discussion regarding these results. Conclusions and future work directions are presented in Section 6. Overall, authors focused on analysis of employee position and process management implications of the achieved results.

2 Social Networks in the Enterprise

2.1 Source Data for Social Network

The data for the social network can be gathered from a vast variety of sources. One of the most often used methods is based on questionnaires and interviews. People are asked to identify the frequency of communication with others and the channel of interaction, e.g. face-to-face meetings, telephone, email, paper letters, etc. [8]. They need to recall their behavior that has taken place over the long time frames in order to capture as much information as possible. However, if the time frame is too long or the information is too detailed, accuracy and reliability of such a method can be very low. It has been proven in [2], [3] that the data based on recall, although widely used, may be much less reliable than the data gathered from observation. Moreover, recall may be better for perceptions of media usage, while the electronic data may be better for measuring actual usage.

Nowadays, when social networking sites have become an essential part of everyday life, questionnaires are no longer needed. These sites allow users to construct an online profile with explicitly defined connections with other users (usually called “friends”) [5], [19]. Since web 2.0 sites like Facebook, LinkedIn, MySpace and many others have been established, they have attracted more than one billion users worldwide. People use social networking sites in their daily lives, to communicate with friends, share interests, organize events, find jobs and many more. However, connections in such systems are self-defined by users, which means that it can be subjective.

Instead of asking people to provide details of their communication, parallel collected data can be used. Recently, people communicate with each other electronically (by email [17], phone [16], or Instant Messenger, etc.), they leave a trace behind them in the form of email server logs, phone billings, IM logs, etc. The more facts of communication between users, the stronger they are connected within the social network. Corporate information systems introduce even more possible sources for social network extraction, like directory services (e.g. Microsoft Active Directory [9]) or ERP systems [20] which allow to analyze not only pure communication but users` business relations as well. Figure 1 presents few of all possible data sources for extraction of the social network.

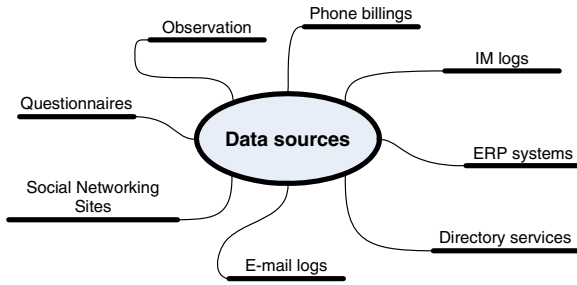


Fig. 1. Data sources for social networks extraction in an enterprise

2.2 Social Network Extraction

Social network extraction should be considered as the most important preliminary step in all further activities on the obtained social network. The final results may differ depending on the approach chosen at this stage. As previously stated, there are many diverse source data for enterprise social network extraction. However, when deliberating enterprise architecture improvement ideas, it should be noted that it is essential to obtain the scheme of organizational structure and to gain at least one type of source data for the company social network extraction.

Overall, the organizational structure should be treated as the snapshot state of the enterprise representing formal dependencies between individual employees. Those dependencies may vary - from simple ones, as functional structure, through matrix structure, up to complex horizontal designs oriented mostly to processes in the enterprise [6]. Paradoxically, such kind of relatively well-defined corporate relations, i.e. the organizational structure can be hardly extracted automatically, because, depending on size and profile, the companies may have no need to maintain full company structure drill-down from board through departments up to a single employee in their IT systems. That is why, it may be necessary to convert organizational structure, taken from official documents, into graph, where nodes are representing employees and vertices – employee-supervisor relations. This was also partly necessary in the analyzed company. Additionally, each node, i.e. each employee, is labeled with the department ID and function what is especially useful in business process analysis. It is also important to have the opportunity to track all the organizational changes during the analyzed period, i.e. employment or dismissals, position or department changes, etc. The company structure, just like real social network, also varies over time and that fact may be crucial in further analysis.

Obviously, the process of the social network extraction may differ due to accessibility of the data. The most common case refers a single-layered social network, in which the network is extracted only from one simple data source, e.g. logs about e-mail communication [4], IM communication [18], etc. An ERP system is the more sophisticated example of social network source and analysis of transaction logs in such case can be valuable for business process improvement [1]. If there are more data sources available, it is worth building a multi-layered social network, in which each layer represents relations inside one channel of communication or common activities [10], [13]. Further social network analysis (SNA) and enterprise architecture improvement considerations (see Sec. 3 and 4) can analyze those layers separately or

flattened with weights assigned to each layer, depending on purpose. It is worth mentioning that the corporate structure can also be used as an additional layer in SNA.

In both cases – a single-layered and multi-layered social network - there is one more consideration: should a social network be limited only to company employees or should also outside connections matter? In the first case, the number of nodes in the social network will be equal or less than the number of nodes in the corporate structure network. In the second case, it is proposed to limit the output social network to company employees, but instead of including outside nodes, one or more additional labels are added to employee nodes informing about count of external connections or count of e-mails sent outside, etc. Such information can be included further in social network analysis as another factor for calculation.

While extracting a social network from one or multiple sources, yet another difficulty may appear – user identification. In a single layer, one user can use multiple aliases [14], [4] and in case of many layers a need to identify the same user across the layers is even greater. The suggested way to start such pre-processing is to use the company structure node list first, because such network has only one occurrence of a single employee. All further layers should be mapped and reduced to the company structure nodes list, because in that case all users will be uniquely identified, as shown in Figure 2. Multiple instances of the same employee in a layer should be reduced to one at this stage. Without proper pre-processing the social network may be hard to analyze.

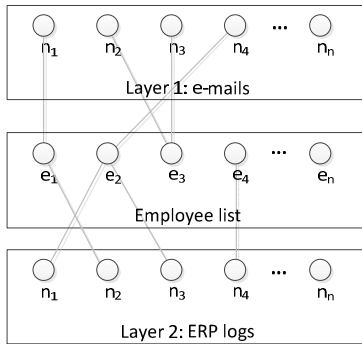


Fig. 2. Mapping social network actors to employee list

There are also special cases in the mapping process, strongly depending on company operating model. For example, if a group of employees uses one login in ERP system or a single e-mail account, such a functional account cannot be simply mapped to a single employee and sometimes additional internal company knowledge is required to perform mapping correctly. In some cases it is even impossible to merge layers containing functional entities with layers including individuals and then each layer could be analyzed separately.

2.3 Multi-agent Architecture of the System

As presented in the previous section, the data used for building the corporate social network may be stored in various, independent systems. That is why the multi-agent architecture is very helpful in the process of obtaining the data and generating the

social network. It is proposed to build an agent architecture, where each agent or agent set is responsible for one source representing one social network layer.

For example, after identification of possible source data location, one set of agents may gather the organizational structure data in Active Directory catalogue, the other will analyze mail logs, while some others will focus on contacts and meetings data stored in the Microsoft Exchange server. The result of the proposed approach will be a multi-layered social network built effectively and independently.

3 The Idea of Corporate Social Network Analysis

The idea of corporate social network analysis consist of application of social network analysis methods (SNA) to social networks built on various company communication and event data. Such analysis can provide an additional information on groups, relations and information flow in a company which may be further used for improving company management in various ways [21]. Also regularities and anomalies in processes, key persons extracted from the social network and many other factors may be found and all these can be considered crucial in finding competitive edge for organizations.

Typically, social network actors are persons, i.e. customers, social networking sites users, employees, etc. What may be interesting is that corporate SNA can consider other type of actors – functional actors, i.e. warehouse workers threat as one entity or even non-human actors, like IT system, depending on data source availability. Thus, there may be different networks built based on the same data source and analyses may be performed using various combinations of entities. That makes corporate SNA task interesting, but, in some aspects, also complex and challenging.

As stated, corporate SNA uses mostly same tools as typical SNA – centrality metrics, clustering, group analysis, etc. However, due to its nature, corporate SNA takes into account also some other company information, such as process definitions and HR information. Conjunction of them can facilitate improvement of company organization and management, however, it also requires development of new combined data analysis methods. Interpretation of SNA results in the organization and especially corporate management changes based on these results should be made very carefully, because their straight implementation may be often risky and more disruptive than helpful for organization. This aspect will be discussed further in Section 5.

4 Experiments

4.1 Company Description

Researchers gained access to the organizational structure data, e-mail logs and (very limited) process definitions in the mid-sized manufacturing company located in Poland. Company employs 300 persons, whereas 1/3 are clerical workers, the rest – laborers. The period analyzed was half a year. The type of organizational structure is functional [6]. However, due to organization operating model and its consequences to organizational structure clarity as well as logs interpretation possibility, only a subset of organization have been chosen for analysis: 49 clerical employees not directly related to manufacturing process. There exists three-level management structure in the selected company part: management board (2 persons), managers (11 persons) and regular employees (36 persons) and they work in twelve different departments.

4.2 General Assumptions

Social network has been build using e-mail logs with the following assumptions:

- 49 internal identities (employees) analyzed (the reasons described above)
- e-mails addressed to self, auto responses (e-mail delivered, out of office etc.) and correspondence with IT systems (ERP and helpdesk system) were omitted
- resulting graph is directed and weighted with the weight of an edge between node i and j is as follows:

$$w_{ij} = \frac{\sum e_{ij}}{\sum e_i} \tag{1}$$

where $\sum e_{ij}$ is the number of e-mails sent by node i to node j and $\sum e_i$ is a total number of e-mails sent by member i . It means that weight w_{ij} focuses on local neighborhood of an employee rather than on global network characteristic.

Because of e-mail logs structure, there was no distinction between *To*, *CC* and *BCC* recipients. The resulting set of data contained 11,816 e-mails in total.

4.3 Social Network Profile

Basic characteristics of resulting social network are shown in Table 1.

Table 1. Characteristic of the social network based on company e-mail interactions

<i>Property</i>	<i>Value</i>
Number of nodes	49
Number of edges	1 018
Average number of edges per node	20.78
Range of incoming edges per node	<9;32>
Range of outgoing edges per node	<2;48>

The histogram of node degrees is shown in Figure 3.

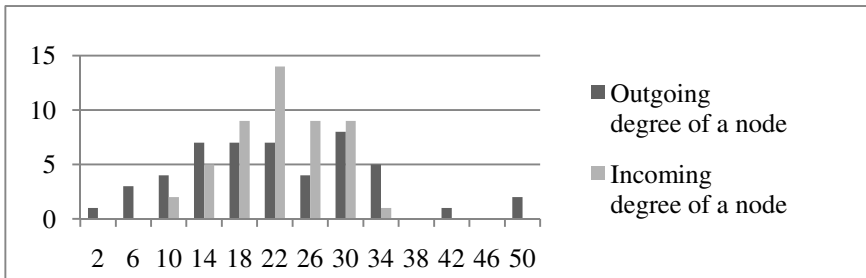


Fig. 3. Histogram of node degrees

4.4 Results

The analysis focused on opportunities to improve enterprise architecture by means of SNA techniques and, in some cases, comparison to organizational structure or company processes definitions.

The first step of social network analysis – entities (node) identification – showed the difference in number of nodes between the employee list (organizational structure) and social network. As concluded, it was caused by an expatriate delegated to work from other company (part-time), however, still using the previous e-mail address. Moreover, the position, which the expatriate occupied, was an important one (head accountant). The lesson learned shows that such situations should be taken into account as a preliminary step before building the social network. The other possible reasons of the same effect: supervisory boards, self-employment, etc.

As regards node degree histogram (Figure 3), six persons were contacting with less than ten recipients and these persons were potentially interesting from the process management point of view. Hence, after acquiring information about processes those employees are involved into, it has been found that the process was carried out in non optimal way in three cases. If these persons would use proper IT system or e-mail instead of paperwork all of those processes may be improved.

Another analysis headed towards finding key persons in each department (partition of social network) and further comparison with the organizational structure. It has been found that among many centrality measures [7], [15], a variant of in-degree centrality $IDC^w(j)$ provides very good results in finding department managers in the social network. It is computed in the following way:

$$IDC^w(j) = \sum_{i=1}^{n-1} w_{ij} \tag{2}$$

where n is the number of all nodes in the social network. In comparison with out-degree centrality, where weight is based not on the number of incoming mails, but on the number of outgoing mails, it has been found, that most managers are rising in in-degree centrality rankings (Figure 4). It means that managers are becoming identifiable not due to their relations to the others, but due to the others' relations to these managers.

What may be interesting, two non-managers have been found in the top ten nodes of the IDC^w rank compared to the organizational structure, but after consultation with the company these two members were also managers at that time: the first person was

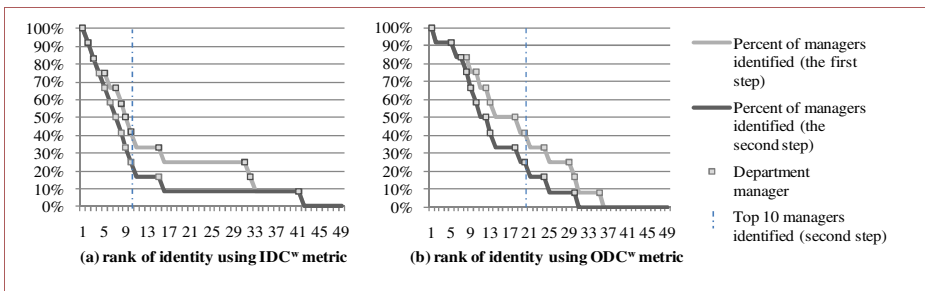


Fig. 4. Managers identified in company social network

the formal assistant manager of the before mentioned part-time expatriate and the second one was the executive temporal manager because the original manager was on long-time sick leave (longer than the analyzed period). It is visible in Figures 4(a) and 4(b), where the first step is based only on the social network compared to the organizational structure and the second step reflects those two special cases.

5 Discussion on Enterprise Improvements

It has been shown that corporate SNA can provide diverse findings. The question is how companies should use the results in a reasonable manner. As previously stated, companies should not base only on SNA, because it is only a support tool. All the results suggesting that some changes can be made should be carefully analyzed and consulted with the management, quality control department, HR department and other authorities involved in the discussed topic. The reason for a such broad discussion is quite simple – some facts may be simply hidden for SNA (not stored in any IT system) and consultations can either confirm SNA results or reveal this latent knowledge. For example, the analysis performed on one available layer – mail server logs - will prove that one company manager is extremely separated from other department managers or even company board. It may suggest that he is alienated from others, what, in general, is undesired on that type of position. Despite of that, further consultations showed, that he prefers direct meetings with co-employees rather than electronic communication and, for that reason, he should not be relocated or dismissed. That situation is also worth deeper analysis, because it may suggest that there is a need to gather information from more sources in that particular company, i.e. PIM appointment lists as a new layer. Another benefit is that this SNA result may also suggest some changes – in case of direct meetings not all decision contents may be written and if there will occur some controversy regarding one of decisions, there would not be a trace to the final decision maker. That is why company management may suggest using e-mail communication to that particular manager in order to improve transparency of decision making process.

The other example worth discussion is the above presented in-degree centrality metric which showed that it is possible to find department managers using SNA. However, why should this metric be considered as the most valuable one? And what decisions should be undertaken by top management if some real managers were found below their subordinates? Or, maybe, if more layers in the same company were analyzed, the rank would differ significantly?

There are some more elemental and easier to interpret findings possible. The example of expatriate revealed that some information may be not properly protected, because internal e-mails were stored outside the corporate infrastructure where security policies may be not on desired level. It may lead to some changes in internal communication, improving overall company security. Nodes degree analysis was also valuable in terms of process management shortening the delays in company information flow.

Despite all the techniques regarding core data analysis that may be very challenging for SNA experts that the real challenge for companies is to properly interpret and make valuable use of the achieved corporate SNA results.

6 Conclusions and Future Work

The analyzed real-world case has proved that corporate social network analysis may be a way to get another point of view on the company. The different channels of communication between employees may be used as a data source to extract corporate social network and results of such network analysis can be used as a valuable decision support tool leading to company architecture improvement. Albeit, as stated in Section 5, the output of this tool should not be used uncritically.

The research undertaken shown that some metrics and techniques used in corporate SNA are more valuable than others, especially in-degree centrality may be considered as an important metric in further corporate network analysis.

Further work on this topic will include in particular: time-oriented dynamic communication analysis, including sliding windows and weights adapted to time [11], comparison of graphs built separately using different layers as well as the merged ones to analyze metric ranks changes and social network surrounding analysis.

Acknowledgments. The work was supported by The Polish Ministry of Science and Higher Education, the development project 2009-11 and the research project 2010-13.

References

1. van der Aalst, W.M.P., Song, M.S.: Mining Social Networks: Uncovering Interaction Patterns in Business Processes. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 244–260. Springer-Verlag, Berlin (2004)
2. Bernard, H.R., Killworth, P., Sailer, L.: Summary of research on informant accuracy in network data and the reverse small world problem. *Connections* 4(2), 11–25 (1981)
3. Bernard, H.R., Killworth, P., Kronenfeld, D., Sailer, L.: The problem of informant accuracy: the validity of retrospective data. *Annual Review of Anthropology* 13, 495–517 (1984)
4. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories (MSR 2006), pp. 137–143. ACM, New York (2006)
5. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* 13(1), article 11 (2007)
6. Daft, R.L.: *Organization Theory and Design*, 10th edn. Cengage Learning
7. Freeman, L.C.: Centrality in social networks conceptual clarification. *Social Networks* 1(3), 215–239 (1978-1979)
8. Haythornthwaite, C., Wellman, B., Mantei, M.: Work relationships and media use: A social network analysis. *Group Decision and Negotiation* 4(3), 193–211 (1995)
9. Iseminger, D.: *Active Directory Service for Microsoft Windows 2000 Technical Reference*. Microsoft Press, Redmond (2000)
10. Kazienko, P., Bródka, P., Musiał, K., Gaworecki, J.: Multi-layered Social Network Creation Based on Bibliographic Data. In: The Second IEEE International Conference on Social Computing, SIN 2010, Symposium on Social Intelligence and Networking, Social-Com 2010, pp. 407–412. IEEE Computer Society Press, Los Alamitos (2010)
11. Kazienko, P., Musiał, K., Zgrzywa, A.: Evaluation of Node Position Based on Email Communication. *Control and Cybernetics* 38(1), 67–86 (2009)
12. Krebs, V.: The Social Life of Routers. *Internet Protocol Journal* 3, 14–25 (2000)

13. Musiał, K., Bródka, P., Kazienko, P., Gaworecki, J.: Extraction of Multi-layered Social Networks from Activity Data. *Journal of Global Information Management* (2011) (in press)
14. Musiał, K., Kazienko, P.: Social Networks on the Internet. In: *World Wide Web* (2011)
15. Musiał, K., Kazienko, P., Bródka, P.: User Position Measures in Social Networks. In: *The Third SNA-KDD Workshop on Social Network Mining and Analysis*. ACM Press, Paris (2009), Article no. 6
16. Onnela, J.-P., Saramaki, J., et al.: Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 7332–7336 (2007)
17. Palus, S., Bródka, P., Kazienko, P.: Evaluation of Organization Structure based on Email Interactions. *International Journal of Knowledge Society Research* 2(1) (2011)
18. Resig, J., Santosh Dawara, C.M.H., Teredesai, A.: Extracting social networks from instant messaging populations. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) Workshop on Link Analysis and Group Detection* (2004)
19. Traud, A.L., Kelsic, E.D., Mucha, P.J., Porter, M.A.: Community structure in online collegiate social networks. *American Physical Society, APS March Meeting*, abstract #H9.013 (2009)
20. Wagle, D.: The case for ERP systems. *The McKinsey Quarterly* 2, 131–138 (1998)
21. Zack, M.H.: Researching organizational systems using social network analysis. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Science* (2000)

Dynamic Workflow Adaptation over Adaptive Infrastructures^{*}

Rafael Tolosana-Calasanz¹, José A. Bañares¹, and Omer F. Rana²

¹ Instituto de Investigación en Ingeniería de Aragón (I3A)
Department of Computer Science and Systems Engineering
University of Zaragoza, Spain
{rafaelt, banares}@unizar.es

² School of Computer Science, Cardiff University, UK
o.f.rana@cs.cardiff.ac.uk

Abstract. There is emerging interest in many scientific disciplines to deal with “dynamic” data, arising from sensors and scientific instruments, which require workflow graphs that can be dynamically adapted – as new data becomes available. Additionally, the elastic nature of many Cloud environments subsequently enable such dynamic workflow graphs to be enacted more efficiently. One of the challenges of scientific workflows is that they must be designed with the needed level of dynamism to take account of the availability of data and the variability of the execution environment, which can be dynamically scaled out based on demand (and budget). In this paper, we present a novel approach for specifying scientific workflows with the two main requirements of: (i) dynamic / adaptive workflow structure well suited for and responsive to change, and (ii) support for large-scale and variable parallelism. We utilise the *superscalar pipeline* as a model of computation and the well-known Montage workflow for illustrating our approach.

Keywords: Workflow Adaptation, Exception Handling, Petri nets.

1 Introduction

There is emerging interest in many scientific disciplines to deal with “dynamic” data. Such data is increasingly large-scale and distributed, arising from sensors, scientific instruments and simulations. Whereas scientific workflow systems have generally been shown to work effectively with file-based data, providing capability within such systems to support streaming-oriented dynamic data becomes a challenge. As observed in [1], focus on *cyber-physical* applications, modelled as a workflow, often need to combine dynamic data from sensors with static data held within file systems. Two distinguishing characteristics of these types of applications are: (i) availability of data may not be known *a priori* – as it is often driven by external physical phenomenon; (ii) a data stream may be delineated by

^{*} This work has been supported by the research project TIN2010-17905, granted by the Spanish Ministry of Education and Science.

user-defined markers – referred to as “landmarks” in [1]. A variety of application domains fall within this area, such as weather forecasting and ocean observation from remote sensors, analysing video and audio feeds, and more recently analysing data from electricity meters to support “Smart (Power) Grids”.

To effectively support this emerging class of scientific applications, it is often necessary to generate workflow specifications¹ that can be dynamically adapted – as new data becomes available. Additionally, the elastic nature of many Cloud environments subsequently enable such dynamic workflow graphs to be enacted more efficiently. One of the challenges of scientific workflows is that they must be designed with the needed level of dynamism to take account of the availability of data and the variability of the execution environment, which can be dynamically scaled out based on demand (and budget). According to [3], Clouds may also be used as accelerators to improve the application time-to-completion, or to handle unexpected situations such as an unanticipated resource downtime, inadequate allocations or unanticipated queuing delays.

On the other hand, as pointed out by [4], whilst existing workflow engines can specify arbitrary parallel programs, they are typically not effective with large and variable parallelism. Conversely, a streaming model innately supports pipeline parallelism, enabling a sequential combination of processing elements to process the stream in sequence, with each stage supporting multiple concurrent operations on the stream. Stages within the pipeline may be a combination of data conversion, filtering, and analysis.

In this paper, we present a novel approach for specifying scientific workflows with the two main requirements of: (i) creating dynamic / adaptive workflow structures well suited for and responsive to change as described by [3], and (ii) support for large-scale and variable parallelism. We utilise the *superscalar pipeline* as a model of computation and the well-known Montage workflow for illustrating our approach. In the pipeline model of computation, a vector of input data elements is streamed into a *sequence* of tasks (or stages): parallelism is achieved as input data elements are processed simultaneously by the pipeline of tasks. In a superscalar pipeline [2], there can be multiple task instances per pipeline stage, provided multiple resources are available. Our model is represented using Reference Nets, a class of Petri-nets which can be dynamically configured.

The rest of the paper is structured as follows. In Section 2, a brief overview of Reference nets is provided. In Section 3, the approach for adaptive workflows is given. Related work is discussed in Section 4 and finally the conclusions are given.

2 Background: Reference Nets

Petri nets have been recognised by their ability to represent parallel or concurrent processes. A Petri net is a graph with two kind of nodes, places and

¹ A scientific workflow can be seen as a high-level specification of a set of tasks and the dependencies among them, which indicate the steps that must be satisfied in order to accomplish a scientific experiment. There are different models of computation available in the literature [2], whilst the data flow model is very common.

transitions, which represent conditions and actions respectively. Places can also contain elements called tokens, which evolve through the places to complete the state representation. The execution of actions require the satisfaction of pre-conditions represented by input arcs going from places to transitions, whereas postconditions are specified by output arcs. In high-level Petri nets nodes are typed representing the type of state for each place, the type of event for each transition, and the type of objects associated with the tokens that flow through the net.

The *Reference net* formalism [5] is a special class of high-level Petri net that uses Java as an inscription language, and extends Petri nets with dynamic net instances, net references, and dynamic transition synchronisation through synchronous channels. Every net element can have associated semantic inscriptions: places can have initialisation expressions, which are evaluated and serve as their initial markings. Transitions can be equipped with a variety of inscriptions, including Java inscriptions, in which the equality operator “=” can be used to influence the binding of variables. The inscription language of Reference nets has been extended to include tuples and comma-separated list of expressions that are enclosed in square brackets. Tuples are useful for storing a group of related values inside a token. The nets can hold two kinds of tokens: valued tokens and tokens which correspond to a reference. New net instances can be created by transitions that carry creation inscriptions, which consist of a variable name, a colon (:), the reserved word *new* and the name of the net.

Transitions of a Petri net are independent of one another and may fire asynchronously. However, net instances can communicate with each other by synchronous channels: they synchronise two transitions which both fire atomically at the same time. Both transitions must agree on the name of the channel and on a set of parameters before they can synchronise. The initiating transition must have a special inscription – called *downlink* – which makes a request to a designated subordinated net. A downlink consists of an expression that must evaluate to a net reference (usually a variable), a colon (:), the name of the channel, and an optional list of arguments. On the other side, the transition must be inscribed with an *uplink*. Channels can also take a list of parameters.

Renew [2] [5] is a Java-based Reference net interpreter and a Reference net graphical modeling tool used in this work. Reference nets have been used for specifying scientific workflows [6], and for deriving performance models of them. Reference nets along with Java inscriptions have been used for implementing a service-oriented workflow engine, DVega [7].

3 Dynamic Workflow Adaptation: The Montage Workflow Case

The Montage workflow provides a representative space science application, typically utilising a large number of images that are stored in distributed archives

² <http://www.renew.de>

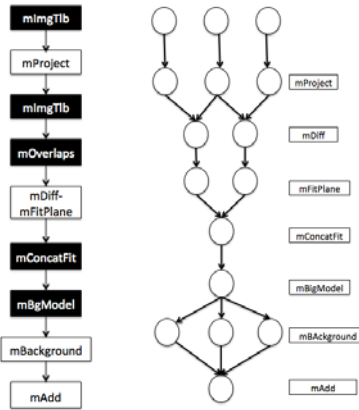


Fig. 1. Montage workflow in terms of the models of computation: (left) superscalar pipeline and (right) data parallelism (for three input files)

and that are, in most cases, remote with respect to the available computational resources. At the end of workflow execution, the outcome is the integration of these astronomical images into a single image mosaic. The processing of these images involve computational and data management challenges, some already addressed by the Pegasus project [8] along with the Montage toolkit [3].

The Montage workflow is generally represented as a Directed Acyclic Graph (DAG) using the data parallelism paradigm. An example of it can be seen in Figure 1 right. In contrast, Figure 1 left shows an alternative representation using the superscalar pipeline paradigm. Multiple data tokens (URLs) representing the (remote) input image files to generate the mosaic are streamed into the pipeline – represented as Reference net tokens in our approach. The main advantage of a superscalar pipeline representation is that the workflow structure remains invariable and is independent of the input. This differs from the Montage DAG approach, in Figure 1 right, whose structure would change depending on the input data files.

One of the challenges of a superscalar pipeline version of Montage, is dealing with the inherent dependencies among some of the data elements within a stream. Some operations have to be applied to a combination of data elements, for instance the *mDiff* operation is used if image files overlap. *mOverlaps* determines which pair of images overlap for a region of the sky, and creates an image metadata table used by *mDiff*. Operations such as *mProject* require all the input data elements of the stream to be synchronised, in order to allow data elements to proceed to the following stage. In the workflow of Figure 1 left, a white background represents a stage where multiple data elements are processed in parallel, whereas a black background represents a synchronizing task that gathers outputs from previous tasks.

³ <http://montage.ipac.caltech.edu/>

All these dependencies among data elements in Montage can be gathered from (i) Montage operations and (ii) *controlled* by means of Reference net constructs. The most elementary of our Reference net patterns are depicted in Figures 4 and 5. In the following subsections, we will describe: (i) how to specify Montage with a superscalar pipeline model of computation and, (ii) how to support workflow adaptation over an adaptive infrastructure (such as Cloud computing).

3.1 Reference Net-Based Pipeline Modelling

Our abstract workflows can have a hierarchical structure consisting of either intermediate nodes or leaf nodes (simple tasks). Elements within a data stream are represented as tokens flowing through the pipeline. Reference nets support different representations [11]: tokens that store remote locations of distributed files, or express structured collections of data (represented as a Java Array). A pipeline is formed by connecting intermediate nodes. Black background tasks in Figure 1 are Montage metadata operations that return files. These provide parallel input argument data for a subsequent White background task. To synchronise tasks within a parallel Montage stage such as *mProject*, we create the Reference net pattern in Figure 2, inspired by [9]. A task starts its enactment in Transition *t3* and finishes in Transition *t4*. In case of exception, Transition *t7* is fired, and the exception is propagated up to Transition *t8* (similar to the propagation behaviour in [6]). The main characteristic of this pattern is the use of double arcs. The double arc in Transition *t2* takes the data element array, Variable *data*, and extracts all its components and puts them in the input place before Transition *t3*. This action implies that Transition *t3* will be fired as many times as elements in the array. The barrier is implemented by the following mechanism in the net: Transition *t6* can only be fired when all the elements in the array finish their execution in Transition *t4*. When Transition *t6* fires, and the results are completely gathered by the Java ArrayList in Transition *t4*, Transition *t5* will be enabled returning the results.

3.2 Workflow Adaptation in the Cloud

To demonstrate workflow adaptation, we use multiple instances of the Montage workflow, with each instance having a different QoS requirement. We use the superscalar pipeline [10] model of computation, whereby multiple data elements can be processed in parallel within a workflow stage as long as there are enough resources available. We consider an abstract and a concrete workflow. An abstract pipeline workflow does not consider task allocation to specific resources (Figure 1 left), whereas a concrete workflow considers surrogate representations of the physical resources (or processing units) for executing the workflow. With multiple surrogate's being mapped to the same physical resource.

To support mapping of abstract workflow tasks to resources, each data element goes through the pipeline of processing units with its own abstract workflow instance. Thus, an abstract workflow instance: (i) indicates the tasks required for the data element and the order in which they must be applied, and (ii)

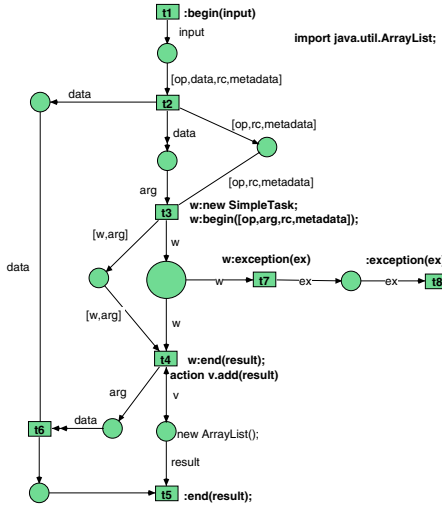


Fig. 2. Distribute and Barrier Reference net-based pattern

acts as the *processing state* for the data element, the number of executed tasks and the number of tasks pending for execution. The mapping mechanism is as follows: a processing unit, on receiving data, attempts to execute as many simple tasks as possible from an abstract workflow instance, subsequently passing the output to another processing unit. Hence, each instance of the abstract Montage workflow represents an independent **stream of data**. As depicted in Figure 3, a node contains a flow isolation mechanism, such as a token bucket, a processing unit, and an Autonomic Data Streaming Service (ADSS). Each token bucket regulates the flow of input to the processing unit, isolates the workflow streams and guarantees that the QoS of each stream is within the pre-defined interval. The processing unit performs the computation by utilising multiple resources in parallel. Depending on the computational requirements of the workflows, resources can be added to or removed from the processing units at runtime on demand (i.e. from a Cloud infrastructure)⁴. Additionally, the flow isolation mechanism prevents workflow data transmissions affecting each other, and the ADSS adapts the transmission rate (due to network congestion, for instance) to prevent data loss.

3.3 Automated Sub-Workflow Replacement

The abstract Montage workflow includes stages with alternative operations, which if chosen may lead to a change in the workflow structure. For instance, executing a parallel-version of *mProject* may result in the creation of multiple task instances which need to be synchronised, for instance by means of the pattern discussed in Figure 2. However, if such a parallel version of *mProject* cannot

⁴ See [11] for a detailed explanation of these models.

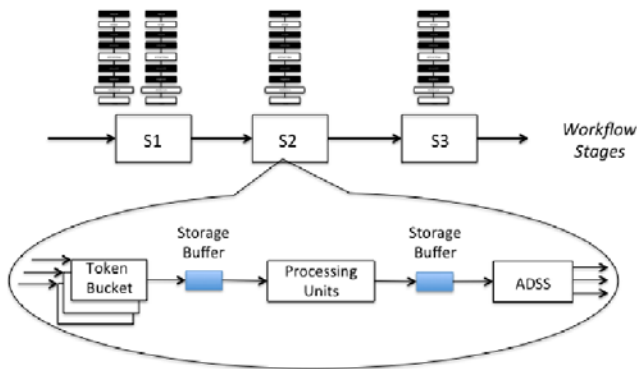


Fig. 3. Workflow System Architecture: the elements of a node and their corresponding remote services it invokes

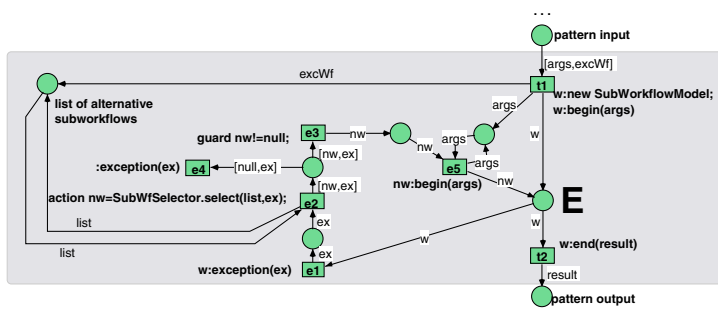


Fig. 4. Intermediate node with exception handling for an abstract workflow specification

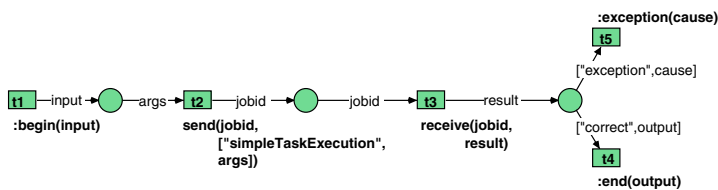


Fig. 5. Leaf workflow node pattern or simple task which ends normally, or abnormally by signaling an exception

be executed due to the lack of suitable resources, an alternative (non-parallel) version may be automatically chosen and executed instead. We consider such an adaption to be similar to an exception handling mechanism when dealing with faults. Using such an approach, each workflow node must have an associated exception handler, which identifies the failure context, and can: (i) propagate the exception up in the hierarchy or (ii) to dynamically replace the signaller (the whole set of descendants of the workflow node) by another set of descendants.

Figure 4 shows the workflow pattern for an intermediate node, identifying alternative sub-workflows that may be chosen based on an exception. An intermediate node receives a token composed by both the arguments required for the enactment (Variable `args`) and a named list of alternative sub-workflows (Variable `excWf`). The firing of transition $t1$ causes the primary sub-workflow to be executed, with normal completion accomplished by the firing of transition $t2$. However, if the workflow execution is to be modified, transition $e1$ is fired instead of $t2$, leading to the primary workflow being retried or an alternative one can be instantiated (based on a user-defined policy [6]).

Figure 5 shows a leaf workflow node. A data element *idata* is received in Transition $t1$. After the firing of Transition $t2$, the task is submitted for execution and completed by the firing of transition $t3$, resulting in output data (*odata*). Execution can complete successfully (firing of transition $t4$) or unsuccessfully (firing of transition $t5$). In both cases, when one of these transition fires, the result will be sent to the next level in the workflow hierarchy. The inscriptions of transitions $t2$ and $t3$ (send and receive, respectively) represent the mapping mechanism between a workflow task and the distributed resource. The Reference nets identified above can be interpreted and executed using the Renew tool. The mapping of workflow tasks to distributed resources can however be undertaken in a number of ways [6].

4 Related Work

Data parallelism has been a natural representation paradigm utilised in scientific workflows for the specification of simple data processing pipelines. However, current workflow languages are remarkably static [4] and the parallelism is specified statically when the workflow graph is generated. The main challenge of modelling data parallelism consists of dealing with a variable number of tasks, whose number may change for each workflow execution. Although it is possible to identify different ways for modelling data parallelism [2], there can be highlighted two main approaches: (i) to build a graph with multiple independent subgraphs; or (ii) to create a graph that contains logic to spawn parallel activities adding looping constructs that scatter and gather threads. Static and Dynamic Graph Rewriting consists in building a graph with multiple independent subgraphs. However, once a graph has started its execution, the amount of parallelism cannot change. Moreover, if there is a lot of parallelism, the graph can become very large and difficult to manage. Pegasus [8] follows this approach by mapping an “abstract workflow” description to an executable form. Pegasus can produce partial executable workflows based on already executed tasks and the currently available Grid resources, instead of a fully specified executable workflow at the beginning. Triana can also [12] dynamically rewrite the graph at run time to map workflow tasks onto the distributed resources.

On the other hand, a way to abstract a dynamic dataset is to pass a vector as input to a task. At run-time, for each element of the vector, a new instance

of the task will be created and started in parallel. JOpera [13] and Kepler [14] follow this approach, and Taverna can also iterate over the cartesian product of the vectors [15]. In spite of its significant power, Kepler, and many other current workflow systems, do not support dynamic parallel execution. However, in [4], a tagged dataflow architecture is proposed as richer execution environment for parallel workflows than current workflow systems.

5 Conclusions

Whilst existing workflow engines can specify arbitrary parallel programs, they are typically not effective with large and variable parallelism, specially in elastic or changing infrastructures. To effectively support this class of scientific applications, it is often necessary to generate workflows whose structure can be dynamically adapted. In this paper, we propose Reference net based patterns for workflow specifications and we utilise a superscalar pipeline model of computation. The modelling expressiveness of Reference nets enables us to represent different runtime adaptive parallelism: (i) workflow structure inherent parallelism that can be modified dynamically at runtime, and (ii) data parallelism due to a variable number of data elements (tokens being flowed through the workflow structure simultaneously and whose data dependencies (if any) can be preserved and controlled by the workflow structure (i.e by a barrier). As the Reference nets proposed in this paper can be interpreted by Renew, our specifications are directly executable. Additionally, Reference nets also allows us to express different data models [1]: tokens that reference files, structured collections of data represented as Java Array and Stream of data represented as abstract workflows that are executed in the pipeline.

References

1. Zinn, D., Hart, Q., McPhillips, T., Ludaescher, B., Simmhan, Y., Giakkoupis, M., Prasanna, V.K.: Towards reliable, performant workflows for streaming-applications on cloud platforms. In: 11st International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011), Newport Beach, USA (2011)
2. Pautasso, C., Alonso, G.: Parallel computing patterns for Grid workflows. In: Proceedings of the HPDC 2006 Workshop on Workflows in Support of Large-Scale Science (WORKS 2006), Paris, France (June 19-23, 2006)
3. Kim, H., el-Khamra, Y., Jha, S., Parashar, M.: An autonomic approach to integrated hpc grid and cloud usage. In: Proceedings of 5 th IEEE International Conference on e-Science (e-Science 2009), Oxford, UK (2009)
4. Abramson, D., Enticott, C., Altintas, I.: Nimrod/k: towards massively parallel dynamic grid workflows. In: Proceedings of the ACM/IEEE Conference on High Performance Computing, SC 2008. IEEE/ACM, Austin, Texas, USA (2008)
5. Kummer, O., Wienberg, F., Duvingneau, M., Schumacher, J., Köhler, M., Moldt, D., Rölke, H., Valk, R.: An Extensible Editor and Simulation Engine for Petri Nets: RENEW. In: Cortadella, J., Reisig, W. (eds.) ICATPN 2004. LNCS, vol. 3099, pp. 484–493. Springer, Heidelberg (2004)

6. Tolosana-Calasanz, R., Bañares, J.A., Rana, O.F., Álvarez, P., Ezpeleta, J., Hoheisel, A.: Adaptive exception handling for scientific workflows. *Concurr. Comput.: Pract. Exper.* 22, 617–642 (2010)
7. Tolosana-Calasanz, R., Bañares, J.A., Álvarez, P., Ezpeleta, J.: Vega: A service-oriented grid workflow management system. In: Meersman, R., Tari, Z. (eds.) *OTM 2007, Part II*. LNCS, vol. 4804, pp. 1516–1523. Springer, Heidelberg (2007)
8. Deelman, E., Mehta, G., Singh, G., Su, M., Vahi, K.: Pegasus: Mapping Large-Scale Workflows to Distributed Resources. In: *Workflows for eScience*, pp. 376–394. Springer, Heidelberg (2007)
9. van der Aalst, W., Moldt, D., Rolke, H.: Pattern based workflow design using reference nets. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) *BPM 2003*. LNCS, vol. 2678, pp. 246–260. Springer, Heidelberg (2003)
10. Pautasso, C., Heinis, T., Alonso, G.: Autonomic execution of Web service compositions. In: 2005 IEEE International Conference on Web Services (ICWS 2005), July 11–15, pp. 435–442. IEEE Computer Society, Orlando (2005)
11. Tolosana-Calasanz, R., Bañares, J.A., Rana, O.F.: Autonomic streaming pipeline for scientific workflows. *Concurr. Comput.: Pract. Exper.* 1–31 (to appear, 2011)
12. Taylor, I., Shields, M., Wang, I., Harrison, A.: The Triana Workflow Environment: Architecture and Applications. In: *Workflows for eScience*, pp. 320–339. Springer, Heidelberg (2007)
13. Pautasso, C., Alonso, G.: The jopera visual composition language. *J. Vis. Lang. Comput.* 16, 119–152 (2005)
14. Ngu, A., Bowers, S., Haasch, N., McPhillips, T., Critchlow, T.: Flexible scientific workflow modeling using frames, templates, and dynamic embedding. *Scientific and Statistical Database Management*, 566–572 (2008)
15. Oinn, T., Greenwood, M., Addis, M., Alpdemir, M.N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M.R., Senger, M., Stevens, R., Wipat, A., Wroe, C.: Taverna: lessons in creating a workflow environment for the life sciences: Research articles. *Concurr. Comput.: Pract. Exper.* 18, 1067–1100 (2006)

ebXML Business-to-Business Integration Based on Semantic Cloud Computing Architecture

Ching-Long Yeh

Department of Computer Science and Engineering, Tatung University,
40 Chungshan N. Td. 3rd Sec., Taipei, 104, Taiwan
chingyeh@cse.ttu.edu.tw

Abstract. ebXML provides an infrastructure for the management of business collaborations between integration partners from discovery of partners, implementation to execution of business collaborations. In this paper, we combine the technologies of Semantic Web and cloud computing to design a multi-agent architecture for the management of business collaborations in ebXML. The agent programs are built upon the RDF management layer consisting of an RDF store and service functions. The execution of business collaborations are carried out by the involved business service agents with the shared states for aligning both ends. In the future the use of ontology in the management of business collaborations and building QoS aware environment based on the Semantic Web require further investigation.

Keywords: Business collaborations, ebXML, Cloud computing, Semantic Web, Multi-agent architecture.

1 Introduction

Electronic business standards, for example, ebXML¹, RosettaNet², have been developed to account for the interoperability among heterogeneous business application systems. The architecture of the standards platforms generally consists of the business collaborations and the message services supporting the higher business interactions. Business collaborations can be specified formally using XML schema, for example, BPSS [1] in ebXML, or in document, for example, PIPs in RosettaNet [2]. RosettaNet PIP is a set of atomic business collaborations which can be used to form wider business scenarios such as supply chain management. BPSS provides an XML schema for business collaboration designer to capture the basic and wider parts of business collaborations in RosettaNet in an XML document.

Business collaborations can be executed based on the BPEL engine by converting the business choreographies in BPSS or RosettaNet PIPs into business orchestrations in BPEL [3][4][5][6][7]. Another approach to executing the business collaborations of BPSS is based on state machine models [9][10]. The platform for executing business

¹ ebxml.org

² rosettanet.org

collaborations must provide a mechanism for managing the QoS requirements as specified in each of the BPSS instances [8][9]. In addition to the run-time phase executing business collaborations, mechanism is also required to manage the creation of business service frontend in each trading partner of the business collaborations during the implementation phase. In addition, in the discovery phase function is desired as well for connecting appropriate trading partners using the same business protocols or can negotiate to each other to become the same. In short, a management mechanism is needed for the automation of creation, discovery, and execution of business collaborations in the ebXML platform.

In ebXML, the Registry and Repository component (ebRR) in the infrastructure plays the role of portal of the platform for conducting the management functions for the implementation and discovery phases [11]. In the run-time phase, each trading partner's frontend manages the execution of business collaborations at one integration end, including monitoring the states of execution and QoS of each transaction in the business collaboration. Thus, it would be difficult to manage the distributed connection of frontend systems for the execution of business collaborations. In this paper, we propose to move the connection of frontend systems to cloud architecture to facilitate the management of business processes during the run-time phase [12].

In the cloud the business collaboration applications are delivered as services over the Internet or referred to as a kind of SaaS (Software as a Service). More specifically, the business collaboration applications in the cloud correspond to the frontend systems in the ebXML technical architecture [11]. They can be connected to the backend application systems using standard protocols SOAP for exchanging structured information. There are various kinds of schema specified in the ebXML specifications, CPP/A, BPSS, ebRR Information Models [11] *etc.* Therefore we use the RDF and OWL used in the Semantic Web as the integration basis to host the ebXML architecture [13].

In Section 2, we give a brief introduction to the relevant background. In Section 3, we describe the architecture of cloud for ebXML. In Section 4, we describe the execution of business collaboration in the cloud. In Section 5, we address several issues which require further investigation. Finally we make conclusion.

2 Technical Architecture of ebXML

ebXML is a service-oriented architecture for conducting business automation. It is an electronic business standard framework developed jointly by international standard organizations, UN/CEFACT and OASIS. It aims at developing a common business platform to enable enterprises of any size in any global region, to conduct business using the Internet [15].

As described previously, ebXML technical architecture supports discovery function to link matched trading partners; trading partners can design their business interaction protocols and then constructs codes accordingly. The ebXML framework consists of the following components to support the above functions.

- **CPP and CPA** languages describe the electronic business capabilities of a trading partner and the agreement of electronic business protocols between them [16].

- **Registry services and information model (RR)** Interfaces for publishing and discovering trading partners' CPPs and ebXML related specifications and model for storing CPPs and specifications [17][18].
- **Business Process Specification Schema (BPSS)** A language for describing the rules of business collaborations [1].
- **Core Component (CC)** Defining the semantics of business documents that are exchanged [19].
- **Message service (MS)** A reliable and secure SOAP-based message exchange service [20].

Based on the framework, the automation of ebXML interoperations between trading partners is carried out in three phases: implementation, discovery and run-time. In the implementation phase, a potential ebXML trading partner, Company A, consults an ebXML registry to acquire the necessary specifications for the implementation of its ebXML system. Company A may implement the system by itself or utilizing third party applications. Company A then registers its electronic business capabilities described in CPP in an ebXML registry in order to be discovered by others. The content of profile consists of specifications of business process and business document which are created using the standard methodology, UMM. UMM is based on the Unified Modeling Language (UML) [21]. A number of production rules are used to convert the resulting models in UML to XML. Therefore both the specifications of business process and business document have their XML schema such that they can be hyper-linked in trading partner's CPPs. All of the above knowledge about ebXML, either in UML or XML, is registered in an ebXML registry for users to access the content.

In the discovery phase, another company, B, can look for potential trading partners and obtain the profile of a potential trading partner, assuming Company A, from ebXML registry. Company B along with the counterpart, Company A, can then carry out a negotiation process to reach an agreement of the contents of their profiles. The agreement between a pair of companies is an XML document that consists of the information about the business processes and business documents used in the processes, and the message services supporting the processes. The negotiation process can be done either automatically or manually, which is outside the specification of ebXML [11].

Both parties hold a copy of the resulting agreement between the pair of companies and then use it to implement their execution codes, called business service interface (BSI). During the run-time phase, the pairs of BSIs play the role of front-end of the back-end processing as shown in Fig. 1 The interactions between the pair of BSIs correspond to the public business processes as described in the agreement. Each time when the CPA is changed or a new one is created, the pairs of BSI need be changed accordingly or a new BSI is created. The task of transforming from specification to executable code may take a lot of human efforts. In this paper, we employ an interpretation approach to relieve the human efforts in the transformation task, as described in Section 4. The resulting BSI after the transformation task is used to carry out the business collaborations as described in the CPA. Thus in this paper we focus on the system design for the implementation and run-time phases. The task of discovery phase is left for future study.

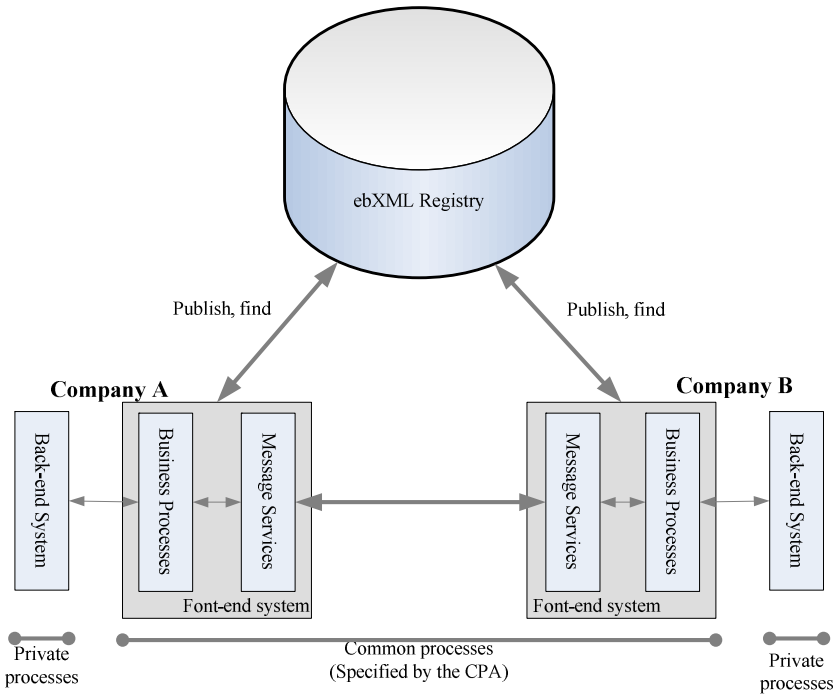


Fig. 1. Conceptual diagram of ebXML technical architecture

3 Agent Architecture for Business Collaborations in the Cloud

The idea of moving business collaborations in ebXML to the cloud can be briefly illustrated by drawing the cloud boundary in the technical architecture as shown in Fig.2. The front-end systems in charge of business collaborations are brought into the cloud. The dialogues happened in the business collaborations are confined within the cloud boundary rather than using the message services across the Internet. Therefore each party of the business collaboration has an agent which collaborates with its counterpart and taking message from and responding message to its corresponding back-end application system.

The cloud architecture can be divided into three layers [11]. The bottom layer is the provider of cloud as utility computing; in the middle is the Service as a Software (SaaS) provider or the cloud user; on the top is the user of the web applications provided by the SaaS provider. The cloud in Fig. 2 is in the middle layer. As mentioned previously in this section, each party has an agent on behalf of it when business collaborations are executed in the cloud. We therefore extend the agent-mediated front-end system we developed previously [9] to accommodate the management and execution of business collaborations.

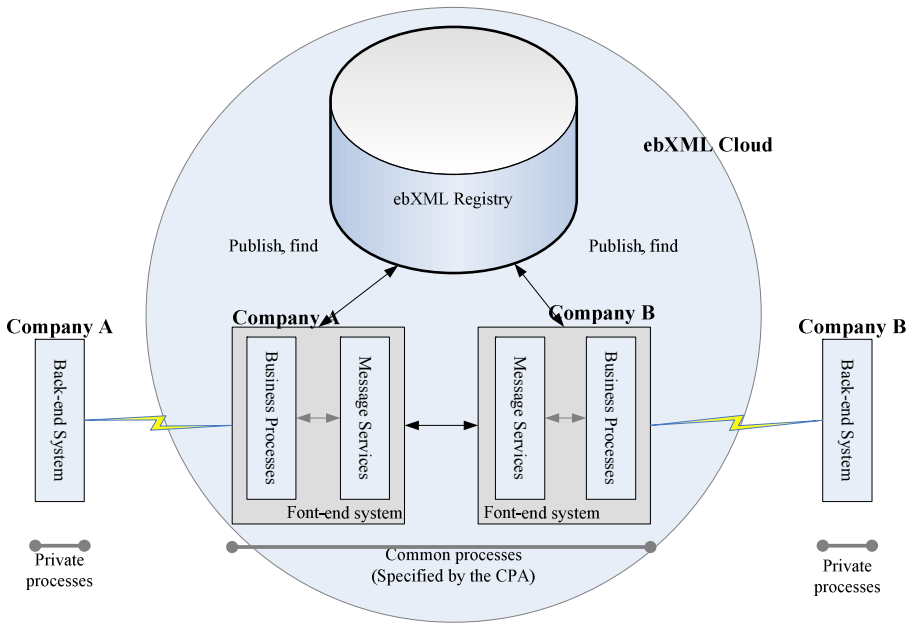


Fig. 2. Business collaboration in the cloud

The development of ebXML applications is divided into three phrases, discovery, implementation, and run-time. In the discovery phase, a pair of potential partners finds out each other through a registry and negotiates to reach an agreement specified in an XML document, CPA. In the implementation phase, the content of CPA is transformed into executable code at both sides which is used in the run-time phase for the execution of the business collaborations. Thus in the agent platform, we create business service agents playing the requesting and responding roles in the business collaboration, respectively.

Having the business service agents built, the system also needs functions for managing the run-time status of the collaboration programs engaging with various counterparts. The management functions include creating a new collaboration issued by the back-end system, modifying or deleting existing collaborations, and querying the status of the running collaborations. In this paper we employ a facilitator agent in agent architecture, OAA [22], for example, to implement the management functions. Each collaboration program is therefore wrapped as a collaboration agent in the agent architecture. This results in agent architecture for ebXML application as shown in Fig. 3.

According to the architecture shown in Fig. 2, the ebXML agent architecture in Fig. 3 corresponds to the cloud boundary shown in Fig. 2. In ebXML, business collaboration consists of business protocol and the message service [20]. The back-end system contacts the Facilitator agent to invoke new business collaboration, which in turn creates the involved Business Service Agents. The Business Service Agents send and receive messages to and from their counterparts. User can invoke the UI agent to access through graphical user interface the services provided by the ebXML agent

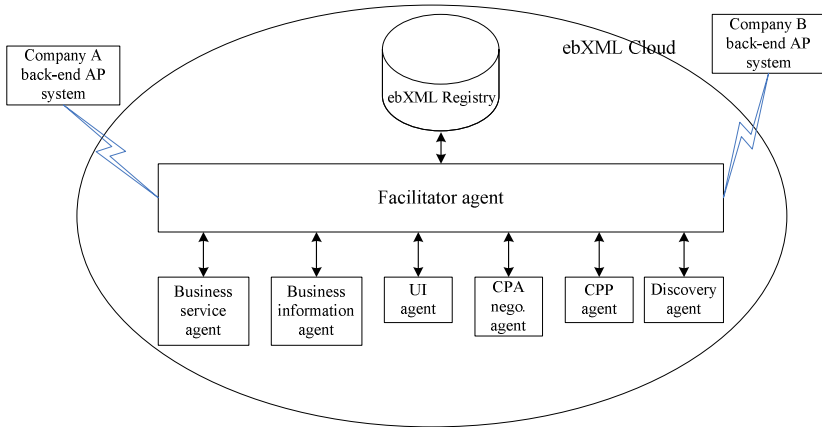


Fig. 3. Agent-mediated architecture upon the cloud for the management and execution of business collaborations in ebXML

platform, for example, managing CPP, looking for trading partners and invoking the business collaborations. The Business Information Agent provides business information in CPA or BPSS for other agents.

In the agent architecture, CPP Agent, CPA Negotiation Agent and Discovery Agent are used in the discovery phase in ebXML application development. The CPP Agent is used to manage the collaboration protocol profile. The CPA Negotiation Agent is used when negotiating with a potential trading partner to achieve an agreement of collaboration protocol. The Discovery Agent is used to find out potential trading partner. In this paper, we focus on the design of the UI, Business Service, and Business Information Agents and leave the rest agents in the architecture in the future study.

4 Execution of Business Collaborations in the Cloud

The business collaborations in BPSS are composed of Business Transactions (BT) in a declarative way. The BTs define the workflow of business document exchange along with parameters for specifying the quality-of-service (QoS) and state alignment [8]. The execution of business collaborations in BPSS has been implemented using the integration technology of Web Service, BPEL [4][5][6][7]. Schönberger *et al.*, further consider the aspect of composing BTs within BC and the realization of QoS to develop finite-state model for the implementation of business collaborations in BPSS [8][9].

In this paper, we employ another important integration technology, the Semantic Web [14], to develop the business-to-business integration in ebXML. First of all, we formalize the integration partners and business collaborations in the ebXML platform by use of the entity-relationship model RDF in the Semantic Web. Second, we use the same entity-relation model to implement the registry services in the ebXML, so that the profiles of integration partners, agreements between integration partners, and artifacts of specifications in ebXML can be managed by the registry service. In this paper, we focus on the first part.

As described in [8][9], the choreography language BPSS supports most of the modeling concepts of business-to-business integration, including the import of business document definitions, specification of flow control of message exchange and roles and relevant QoS properties. They divide the choreography into abstract and lower levels. In the lower level, termed micro-choreography, are business transactions that deal with the flow of documents along with acknowledgement signals to and from a pair of integration partners. Then in the abstract level specifies the transition from one state to another where a state represents the business transactions at lower level. In our previous work [10], we transform the business collaboration specified in a BPSS instance into a two-level finite state machine. The low-level finite-state machine performs the flow of documents and the associated acknowledgement signals between the integration partners in the business transaction. The high-level finite state machine specifies the transition of states as business transactions in the business collaborations.

The execution of a BT can be defined as the alignment of state between integration partners. It includes the exchange of business documents, business signals, and fulfillment of QoS requirements. Thus the requesting and responding agents executing a BT must cover all the items listed above. Also the BT agents must be configurable in the QoS need to satisfy the requirements of various scenarios.

When executing business collaboration, each integration partner shares its own states consistently with its counterparts. The shared states change as a BT is successfully executed. In distributed environments [8], each integration partner owns its part of the share states and synchronizes with its counterpart by sending/receiving business documents and associated signals. In this paper, the cloud computing architecture hosts the agent-mediated business collaborations. We therefore create a shared state

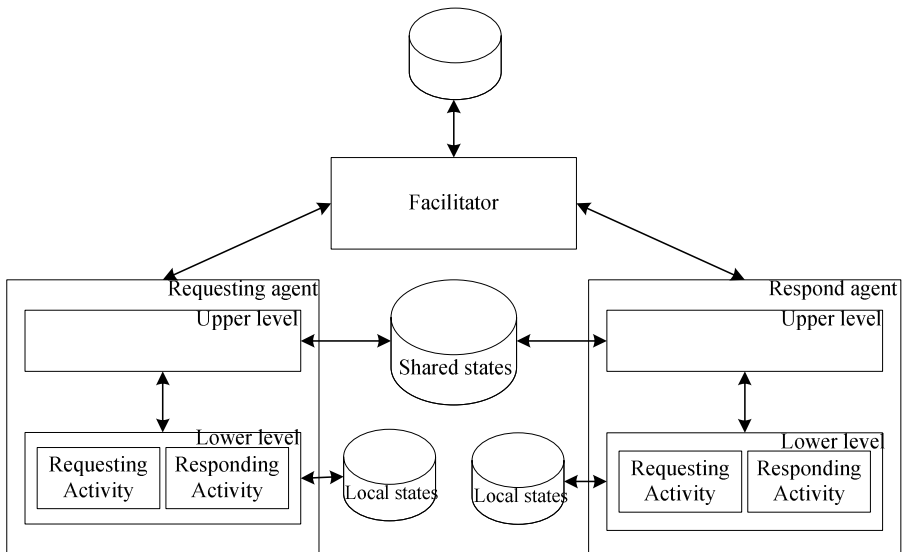


Fig. 4. Business service agents with shared states and their own local states

for the newly invoked business collaboration; each of the integration partner's agents can access and annotate the shared area to indicate the progress of business collaboration from their own aspects. Each integration partner's agent maintains its own state for in the micro-choreography within the business collaboration.

Thus, based on the two-level choreography model, when an integration partner's backend application system call for a business collaboration through its own interface agent as shown in Fig. 3., the facilitator creates the involved business service agents according to the content in the CPA as shown, for example, in Fig. 4. The workflow program embedded in the business service agent is divided into the upper-level and the lower level parts. The former is in charge of the state change and alignment between the collaborating agents; the latter handles the exchange of business documents and the associated business signals to make the change of states in the upper-layer.

5 Discussions and Future Work

In this paper we present the design of a cloud computing architecture for the execution of business collaborations in ebXML. There are still several issues which require further investigation.

First of all as mentioned previously we employ the Semantic Web technology as the integration basis for the design of the cloud computing architecture. RDF is the integration language used in the Semantic Web. In the cloud architecture, an RDF management layer to support all the activities happened in discovery, implementation and run-time phases of ebXML platform. The RDF management layer consists of an RDF store and associated management services. Current the technology for building large RDF triple stores is mature³. Also research on building RDF store based on open cloud architecture emerges recently as well [23]. Thus we have sufficient choices of technology to build up the RDF management layer.

The ebXML Registry Services (ebRR) play the essential role in the discovery and implementation phases. The ebRR already has defined in the specification the information mode and service interfaces. Thus we will further investigate realizing the schema defined in the ebRR specification using the schema language in the Semantic Web, OWL. The service interfaces defined in the ebRR are then implemented using the RDF query language SPARQL. As for the runtime phase, we develop schema in OWL for the implementation of shared states between the collaborative agents, local state of each business service agent and the store associated with the facilitator as shown in Fig. 4.

Second, the agent architecture must be QoS aware. In ebBP, the workflow in business collaboration can be divided into upper and lower level. The former is concerned with the change of states shared between collaborative parties which are control by the exchange of documents and QoS attributes in the latter. Schönberger, *et al.*[9] have summarized the list of QoS attributes related to business documents exchanged, business transactions, business activity and business transaction activities. They further employ state machine model to analyze the security and reliability problems related to the QoS attributes in ebBP. In this paper, the architecture for business

³ See LargeTripleStores (<http://www.w3.org/wiki/LargeTripleStores>) for details.

collaboration consists of cloud computing, Semantic Web, and multi-agent. We therefore resort to the security and reliability issues related to these technologies.

6 Conclusions

In this paper, we try to combine the technologies of Semantic Web and cloud computing to design a multi-agent architecture for the execution of business collaborations in ebXML. An RDF management layer consisting of an RDF store and RDF management functions is built upon the cloud computing infrastructure. The workflow of business collaborations are then developed based on the layer. By representing the transitions of states in the higher level and flow of business documents and signals in the lower level of business collaboration, we can implement the workflow program in a declarative way to facilitate the management of business collaborations in the runtime phase of ebXML. Moreover, we will enhance the security and reliability of QoS attributes based on the security and trust framework in the Semantic Web.

Acknowledgements

Research for this paper was financed by National Science Council, Taiwan, under contract NSC 99-2221-E-036-039.

References

1. OASIS: ebXML Business Process Specification Schema Technical Specification, 2nd edn., OASIS (December 2006)
2. Damodaran, S.: B2B Integration over the Internet with XML: RosettaNet Successes and Challenges. In: Proc. of WWW 2004, Alternate Track Papers and Posters, pp. 188–195. ACM Press, New York (2004)
3. Kim, J.H.: Christian Huemer: From an ebXML BPSS Choreography to a BPEL-based Implementation. *SIGecom Exchanges* 5(2), 1–11 (2004)
4. Khalaf, R.: From RosettaNet PIPs to BPEL Processes: A Three Level Approach for Business Protocols. *Data Knowl. Eng.* 61(1), 23–38 (2007)
5. Schönberger, A., Pflügler, C., Wirtz, G.: Translating Shared State Based ebXML BPSS Models to WS-BPEL. *Int. J. of Business Intelligence and Data Mining* 5(4), 398–442 (2010)
6. Schönberger, A., Wirtz, G.: Realising RosettaNet PIP Compositions as Web Service Orchestrations - A Case Study. In: 2006 International Conference on E-Learning, E-Business, Enterprise Information Systems, E-Government, & Outsourcing, pp. 141–147. CSREA Press (2006)
7. Heravi, B.R., Razzasi, M.: Utilising WS-BPEL Business Processes through ebXML BPSS. In: International Conference for Internet Technology and Secured Transactions, London (2007)
8. Schönberger, A., Wirtz, G.: Using Variable Communication Technologies for Realizing Business Collaborations. In: 2009 IEEE World Congress on Services, Part II, pp. 132–139. IEEE Computer Society, Los Alamitos (2009)

9. Schönberger, A., Wirtz, G., Huemer, C., Zapletal, M.: A Composable, QoS-Aware and Web Services-Based Execution Model for ebXML BPSS Business Transactions. In: 6th World Congress on Services, pp. 229–236. IEEE Computer Society, Los Alamitos (2010)
10. Yeh, C.L., Lu, L.C., Wu, S.S.: Design and Implementation of Declarative Agent-Based Front-End System for ebXML. *Journal of e-Business* 12(1) (2010)
11. OASIS: ebXML Technical Architecture Specification v1.0.4 (2001)
12. Armbrust, M., et al.: Above the Cloud: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28, EECS, UC Berkeley (2009)
13. Hendler, J.A.: Agents and the Semantic Web. *IEEE Intelligent Systems* 16(1), 30–37 (2001)
14. Beckett, D. (ed.): RDF/XML Syntax Specification (revised). W3C Recommendation (2004)
15. Gibb, B., Damodaran, S.: ebXML: Concepts and Application. Wiley, Chichester (2003)
16. OASIS: Collaboration-Protocol Profile and Agreement Specification v2.0 (2002)
17. OASIS: OASIS ebXML Registry Information Model Version 2.0 (2002)
18. OASIS: OASIS/ebXML Registry Services Specification v2.0 (2002)
19. UN/CEFACT: Core Components Technical Specification – Part 8 of the ebXML Framework, Version 2.01 (2003)
20. OASIS: Message Service Specification Version 2.0 (2002)
21. Booch, G., Rumbaugh, J., Jacobson, I.: *The Unified Modeling Language User Guide*. Addison-Wesley, London (1999)
22. Martin, D.L., Cheyer, A.J., Moran, D.B.: The Open Agent Architecture: a Framework for Building Distributed Software Systems. *Applied Artificial Intelligence: An International Journal* 13(1-2), 91–128 (1999)
23. Newman, A., Hunter, J., Li, Y.F., Bouton, C., Davis, M.: A Scale-Out RDF Molecule Store for Distributed Processing of Biomedical Data Semantic Web for Health Care and Life Sciences. In: Workshop WWW 2008, Beijing, China (2008)

SWAV: Semantics-Based Workflows for Automatic Video Analysis*

Gayathri Nadarajan, Yun-Heh Chen-Burger, and Robert B. Fisher

School of Informatics, University of Edinburgh, U.K.
gaya.n@ed.ac.uk, {jessicac,rbf}@inf.ed.ac.uk

Abstract. This paper outlines the SWAV system – Semantics-based Workflows for Automatic Video Analysis. SWAV utilises ontologies and planning as core technologies to gear the composition and execution of video processing workflows. It is tailored for users without image processing expertise who have specific goals (tasks) and restrictions on these goals but not the ability to choose appropriate video processing software to solve their goals. An evaluation on a set of ecological videos has indicated that SWAV: 1) is more time-efficient at solving video classification tasks than manual processing; 2) is more adaptable in response to changes in user requests (task restrictions and video descriptions) than modifying existing image processing programs; and 3) assists the user in selecting optimal solutions by providing recommended descriptions.

Keywords: semantics based workflows, ontologies, HTN planning, requirements based virtual workflow system, intelligent video processing.

1 Introduction

The field of video analysis is becoming more and more important with the increasing size of real-time data that need to be processed today. The pervasiveness of video data, *e.g.* satellite images, surveillance videos and environmental monitoring recordings, has triggered the need for more efficient means to analyse them than just traditional manual means. At present, analysing them is a tedious task as it requires either a large amount of manual processing time and/or highly specialised computational tools. The use of computational tools would speed up this process considerably, however, users almost always do not have access to such tools nor possess the technical expertise to implement or use them.

To provide a context, consider videos of underwater life available to marine biologists. Among the tasks conducted are video filtering, object detection and counting. The filtering involves removing videos that are unusable, *e.g.* those that are too dark or too bright as they are uninteresting for further analysis. The detection would include distinguishing objects of interest, such as fish, and further, these objects are counted for statistical purposes. Later on, they may also want to classify the fish according to their species type. Hence there is a

* This research was funded by European Commission FP7 grant 257024, in the Fish4Knowledge project (www.fish4knowledge.eu).

range of tasks that the user is interested in. Manual analysis involves observing the video clip, pausing the video to take notes and repeating the process until the task is complete. In order to assist users without image processing expertise to conduct **video and image processing (VIP)** tasks in an efficient manner, a suitable form of *automated* assistance should be provided. For this purpose a combination of computer vision methods, workflow and planning technologies and semantics-based approaches were investigated. First a hybrid workflow composition framework is outlined (Section 3), followed by the core components, the VIP ontology (Section 4), the VIP components (Section 5) and the planner (Section 6). The integrated system (SWAV) is then evaluated for efficiency, adaptability and learnability (Section 7).

2 Related Work

E-Science and Cloud workflow systems have emerged as forerunners in providing a specialised environment to simplify the programming effort required by scientists to orchestrate a computational science experiment. Therefore, Cloud-enabled systems must facilitate the composition of multiple resources, and provide mechanisms for creating and enacting these resources in a distributed manner. This requires means for *composing* and executing complex workflows, which has attracted considerable effort especially within the workflow community.

Major workflow systems include Pegasus, Triana, Taverna and Kepler [5]. Pegasus consists of a mapping and an execution engine. The mapping engine maps abstract workflows to its concrete (executable) form. The abstract workflows may be defined directly by application developers (workflow experts), semi-automatically or constructed with the assistance from a workflow editor. Triana is graphical workflow system that has been used for text, speech and image processing tasks. Workflows are created by drag-and-drop and sent for execution or saved. Workflow manipulation is handled manually by the user.

Taverna facilitates workflows for bioinformaticians who have a deep knowledge of the scientific functionality of the resources they want to link together, but limited expertise in programming and middleware technicalities. Workflow construction is placed in the hands of the user who is a domain expert. Kepler is a Java-based workflow system that can model complex computations. Users compose workflows using its graphical user interface.

Pegasus provides assistance to compose workflows using a system that can analyse, verify and correct partial workflows specified by the user. This work, in contrast, aims to automatically or interactively compose workflows from scratch. Triana, Taverna and Kepler provide good graphical interfaces, however, require the user to have domain expertise to compose and manipulate the workflows. To summarise, the limitations of existing workflow initiatives include 1) no provision of automated support in constructing workflows; 2) not tailored to react to changes in user goals and preferences; 3) unable to improve performance autonomously (or with user involvement) in an incremental manner according to specified goals; and 4) no mature integration with ontologies that would allow for more powerful representation and reasoning abilities. This work seeks to address

some of these vital research gaps, by designing a framework that incorporates workflow technology with planning, ontologies and computer vision tools.

3 Three-Layered Workflow Framework

SWAV was implemented based on a hybrid semantics-based workflow composition method within a three-layered framework (Fig. 1). It distinguishes three different levels of abstraction through the design, workflow and processing layers.

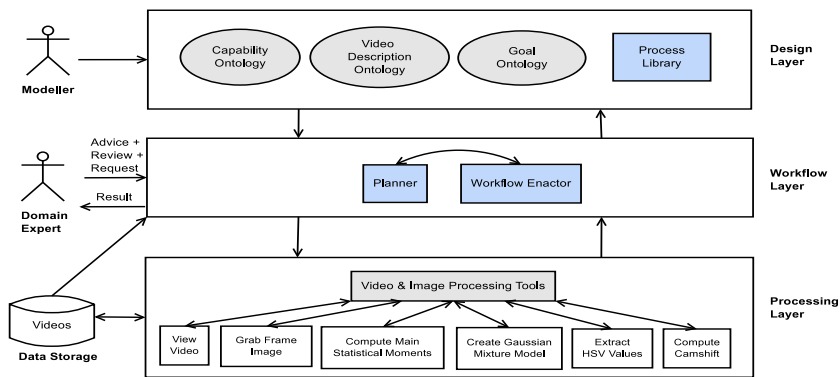


Fig. 1. Hybrid workflow composition framework for video processing with three abstraction levels. The core technologies are ontologies, planning and process modelling.

The design layer contains components that describe the VIP tasks, information about the video, image processing tools and processes to be carried out in the system. These are represented using a modular ontology and a process library. A modeller is able to manipulate the components of this layer, for example populate the process library and modify the ontologies. Typically the modeller is trained in conceptual modelling and has knowledge in the application domain, but not necessarily. Knowledge about VIP tools, user-defined goals and domain descriptions are organised qualitatively and defined declaratively in this layer using ontologies (Section 4), allowing for versatility, rich representation and semantic interpretation. The process library contains the code for the primitive VIP tasks and methods available to the system. These are known as the process models. A primitive task is one that can be directly performed by a VIP tool, while a method is decomposed into primitive and non primitive tasks.

The workflow layer is the main interface between the user and the system. It also acts as an intermediary between the design and processing layers. The workflow enactor ensures the smooth interaction between the components, access to and from various resources such as raw data, VIP toolset, and communication with the user. The main reasoning component is an execution-enhanced planner that is responsible for transforming the high level user requests into low level video processing solutions. More details will be provided in Section 6.

The processing layer consists of a set of VIP tools that can perform various image processing functions. Some examples can be seen in Fig. 4. The functions of these tools are represented in the capability ontology in the design layer. Once a tool has been selected by the planner, it is applied to the video directly. The final result is passed back to the workflow layer for output and evaluation. The derivation methodology of the VIP components will be described in Section 5.

4 VIP Ontology

A pragmatic ontology was required to model the video and image processing (VIP) field so that it can be used for domain description and understanding, as well as inference. The ontology should describe the domain knowledge and support reasoning tasks, while being reasonably independent from the system. The principles adopted for the ontology construction included simplicity, conciseness and appropriate categorisation. For this reason, several aspects of the VIP field were highlighted. These were identified as *goal*, *video description* and *capability*. These aspects were motivated by the context of their use within a planning system that requires the goal and initial domain state model (which includes the initial video description) and also a performance-based selection of operators. The VIP ontology was modularised into three independent ontologies¹. Each ontology holds a vocabulary of classes of things that it represents and the relationships between them. Among the possible domain knowledge representations, ontologies present a number of advantages, the most important being that they provide a formal framework for supporting explicit, machine-processable semantics definition, and they enable the derivation of implicit knowledge through automated inference. A system with full ontological integration has several advantages. It allows for i) cross-checking between ontologies; ii) addition of new concepts into the system; and iii) discovery of new knowledge within the system.

The **goal ontology** contains the high level video processing tasks (goals) and constraints that are communicated by the user to the system. It contains typical goals or classes of VIP tasks such as “Detection”, “Classification”, “Segmentation” and “Compression”. Under each goal umbrella specific instances of goals can be found, such as “classify_fish_green_chromis” and “detect_presence_coral”. Constraints are criteria that give additional restrictions to the goal. These include qualifiers to indicate user preferences such as speed of processing, CPU memory used, reliability of result, and accuracy of detection.

The **video description ontology** contains the concepts and relationships that describe the images and videos, such as the lighting conditions, colour information, position, orientation as well as spatial and temporal aspects. Hence, qualitative concepts such as “bright” (high luminosity) and “blur” (low clearness) could be used to describe the input video. The constraints and video description together constitute the domain description. Based on the goal and initial domain information provided by the user, the goal and video description ontologies are used to formulate the input to the planner.

¹ Visual and formal descriptions of the ontology can be found in <http://homepages.inf.ed.ac.uk/gnadaraj/phd/ontologies>.

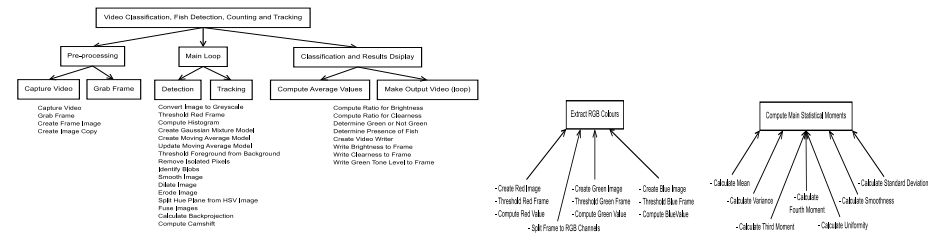
The **capability ontology** contains the classes of VIP tasks and tools that can perform these tasks. Additionally, it organises them hierarchically, links the tasks to the tools and relates the tools with performance measures. Each task is associated with one or more tools (operators). A tool is a software component that can perform a VIP task independently given some input values, or a technique within an integrated vision library that may be invoked with given parameters. This ontology will be used directly by the planner in order to identify the tools that will be used to solve the task. The performance level of the tools are also tied to applicable criteria, namely these criteria refer to the domain information (video description and/or constraints). For instance, “Create Gaussian Background Model” is the best tool to perform a background model creation when the clearness level and the speed of movement are both high.

The main roles of the ontologies include guiding user and workflow for requirement retrieval, assisting image processing-naive users in decision-making by providing recommended descriptions and checking for consistencies.

5 VIP Components

The VIP components constitute the tools or operators of the workflows composed and executed. They are represented in the process library as primitive processes and the capability ontology as VIP tools. Typically, a VIP task is solved by writing a program that is compiled into an executable which can be run on an input video. However, having just one executable would only work on one task or a small subset of tasks. In order to construct such programs automatically, executables of a lower level of granularity would be required.

Generally image processing experts develop a single precompiled VIP tool, or executable that could work on one or a few similar videos for a VIP task. Often this single executable is modified manually before compiled and executed on a different type of video that requires different algorithms. Using the workflow approach designed for this work, such manual modifications are no longer required, as appropriate algorithms according to user and domain information could be selected automatically. Hence a multiple executable approach was devised for



(a) Top-down approach to identify some VIP operators for video classification, fish detection, counting and tracking task. (b) Bottom-up refinement to derive the executables ‘Extract RGB Colours’ and ‘Compute Main Statistical Moments’.

Fig. 2. Top-down and bottom-up combination for deriving VIP components

processing a range of VIP tasks using a selection of VIP tools. This multiple executable system is intended to provide the basis for a modular and reusable way to solve VIP tasks.

A combined top-down and bottom-up methodology was applied to derive a multiple executable system for video classification, fish detection, counting and tracking using single executable OpenCV programs designed for a variety of videos [4]. First, the program code was inspected thoroughly and tasks were broken down in a top-down manner (Fig. 2(a)). This involved breaking down the steps used in solving the task into meaningful blocks or components. Each function call and arithmetic operation was regarded as a primitive task. This exercise yielded 85 unique primitive processes in the process library that were encoded as operators in the capability ontology. When run on a one-minute clip containing 300 frames, 69,011 steps or operator invocations were produced. The bottom level tasks or operators were too fine grained and did not provide a manageable level to work with. They were also too technical for an image processing-naïve user to comprehend and make decisions upon.

Subsequently, the bottom level tasks were grouped by procedure to provide a coarser level of granularity that was more manageable. This involved grouping the bottom level processes (primitive tasks) by procedure. For the most part, the primitive tasks were grouped to represent the subtask one level immediately above them. This exercise yielded 30 operators, termed as *independent executables*, that were much more manageable to work with. This methodology has been used effectively to accomplish the derivation of the VIP components.

The advantage of this bottom-up refinement approach has led to the identification of modules that could be reused for most video processing tasks. In addition, the executables provided a more intuitive representation of the video/image processing tasks than their primitive level counterparts. For instance, in Fig. 2(b), the independent executable “Compute Main Statistical Moments” which was derived by merging seven primitive tasks is a more compact and concise concept to represent a subtask to compute the mean, standard deviation and other statistical moments of an image. With this reduction of almost threefold in the number of operators from 85 to 30, a sample run on the same one-minute clip of 300 frames tested on the operators from the top-down approach now yielded 8706 execution steps, a reduction of almost eightfold in the number of steps [3].

6 Workflow Enactor and Planner

The workflow enactor plays the important role of orchestrating the flow of processing within the system (see Fig. 3). First it reads in the user request in textual form (use selects from a list of options). Next it consults the goal and video description ontologies to formulate the input that is then fed to the planner. When the planner, with the assistance of the process library and capability ontology, returns the final solution plan, the enactor prompts the user for further action. The user has access to the final result of the video processing task textually and visually (step 2), has the choice to rerun the same task on the same video but with modifications to the domain information (step 3), rate the quality of

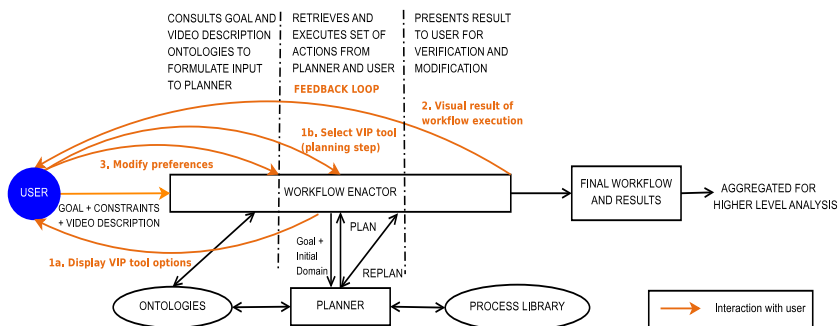


Fig. 3. Overview of interaction between the user, workflow and other system components when planning in semi-automatic (interactive) mode

the result or perform another task. The composed workflow is saved in a script file that can be invoked easily off-line. By being able to view the result of each solution with changes to the domain information, the user can assess the quality of the solution produced. This feedback mechanism could be used as a basis for improving the overall performance of the system as verifying the quality of the video processing solutions automatically is not a trivial task.

The planner acts as the “brain” of the system, which translates the high level user request into low level VIP steps. Adopting the principle that VIP tasks can be solved using a hierarchical decomposition approach, a Hierarchical Task Network (HTN) planner was implemented to realise this. This principle states that a task (goal) may be achieved by performing a set of primitive or non primitive subtasks, each non primitive subtask is further decomposed recursively until primitive tasks are reached. A primitive task could be performed directly by a VIP tool (operator). The role of the planning algorithm is to select the optimal set of VIP tools to achieve a given task. In HTN planning, the goal, initial state, and a set of methods are provided to the system. The methods encode the decomposition of known tasks. For example, video classification according to brightness, clearness and green tone levels may be achieved by first preprocessing the video, followed by computing the average values for the attributes to be classified. Computing the average values for the attributes involves computing the brightness, clearness and green tone levels in each frame image accumulatively. These best known practices adopted by image processing experts or *heuristics* are included as methods in the process library. In an HTN planner, the search space is reduced greatly because only the subtasks that are applicable to solve a current task are considered as nodes for further expansion. The set of options are reduced as planning progresses as only those options that match the preconditions for a subtask (either primitive processes or methods) are selected as valid choices. HTN planners are very efficient as a result of this.

The planner is able to plan in automatic and semi-automatic (interactive) modes. In the semi-automatic mode, it presents to the user all the available VIP tools that can perform a specific primitive task along with their recommended

descriptions whenever more than one tool is available to solve the task. The user can make an informed decision based on these descriptions, making it an informative and interactive tool. In this fashion, the user is given some level of control during the planning phase.

7 Evaluation

30 videos originating from an ecological source via the Ecogrid project, Taiwan [1] were used for evaluating the overall approach. Interesting characteristics of marine life such as fish and coral can be extracted from the videos by performing analysis such as classification, detection, counting and tracking. The videos were taken in an uncontrolled open sea environment where the degree of luminosity and water flow may vary depending upon the weather and the time of the day. The water may also have varying degrees of clearness and cleanness. In addition, the lighting conditions change very slowly, the camera and the background are fixed and images are degraded by a blocking effect due to the compression. Taking into consideration factors such as diversity in user requirements, variety in the quality of the videos (*e.g.* lighting conditions, object movement) and vastness of the data made available, three hypotheses were formulated:

1. Automated support could be provided for users without image processing expertise to perform VIP tasks in a *time-efficient* manner using SWAV without loss of accuracy in the quality of the solutions produced.
2. Constructing VIP solutions using multiple VIP executables employed by SWAV is more *flexible* and *adaptable* towards changing users needs than modifying single executable programs.
3. The SWAV's mechanism to compose and execute workflows for VIP tasks helps the user *learn* the processes involved in constructing optimal solutions.

To test the first hypothesis (efficiency), the task completion time of performing video classification according to brightness, clearness and green tone levels using SWAV (automatic tool) was compared to the task completion time of the classification task conducted manually. Eight participants from a variety of backgrounds, none of whom possessed image processing expertise were selected as subjects, including an ecologist and a marine biologist.

Statistical hypothesis testing using the *t*-distribution [2] was conducted to measure the dependencies between the results obtained for the times taken to conduct automatic and manual processing. For this sample set, the two sample dependent *t*-test was performed to determine the *t* value and its corresponding *p* value in order to accept or reject the null hypothesis. A significance level of $p < 0.05$ was taken as an acceptable condition to reject the null hypothesis. Using the values from Table 1, *t* was computed using Equation 1 below:

$$\bar{d}_e = -36.83 \quad \sigma_{de} = 9.12 \quad n = 8$$

$$t = \frac{\bar{d}_e}{\sqrt{\sigma_{de}^2/n}} = \frac{36.86}{\sqrt{9.12^2/8}} = -11.43 \quad (1)$$

Table 1. Time and accuracy of automatic (SWAV) versus manual processing, and their differences for video classification according to brightness, clearness and green tone levels using 30 Ecogrid videos.

Subject	Automatic (SWAV)		Manual		Difference	
	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	d_e	d_a
1	2.12	61.11	47.90	76.19	-45.78	-15.08
2	2.13	61.11	39.65	53.33	-37.52	7.78
3	2.09	61.11	40.12	25.00	-38.03	36.11
4	2.06	61.11	45.33	87.50	-43.28	-26.39
5	2.13	61.11	35.02	52.38	-32.89	8.73
6	2.14	61.11	48.25	80.00	-46.11	-18.89
7	2.06	61.11	37.20	66.67	-35.14	-5.56
8	2.02	61.11	17.95	52.78	-15.93	8.33
Average	2.09	61.11	38.93	61.73	-36.83	-0.62

where n is the sample size, \bar{d}_e is the mean of the differences between the manual and automatic times and σ_{de} is the standard deviation of this mean. Based on the values of t and n , a significance level was computed. The degree of freedom was set to 7 ($n - 1$). A value of $t(7) = -11.43$ corresponds to a significance level of $p \ll 0.0001$. Therefore the efficiency of automatic processing using SWAV is significantly higher than the efficiency of manual processing. A similar statistical testing was conducted for accuracy, where there was no significant difference in the accuracies of the two methods. Hence the efficiency of automatic processing is significantly higher than manual processing without loss of accuracy.

The second hypothesis (adaptability) was tested using two subjects, an image processing expert and a workflow modeller, to make changes to the system available to them to perform fish detection and counting task on a video when domain descriptions (user preferences) change. Both have access to the same set of VIP tools; the former has an OpenCV program with available image processing algorithms written as functions and the latter in the form of multiple executables within a planning and ontology-enhanced workflow context, as defined in the SWAV tool. The time taken to make the appropriate modifications for six types of changes are contained in Table 2.

Statistical hypothesis testing using the t -distribution was conducted to measure the dependencies between the results obtained for the times taken to make changes to the OpenCV program and the SWAV tool. The significant level of $p \ll 0.05$ was obtained, proving that the workflow tool is faster to adapt to changes in domain descriptions than the image processing program.

The third hypothesis (user learnability) was conducted using the same eight subjects from the first experiment. Each was given 14 pairs of videos to work with, to perform fish detection and counting task using SWAV. Each pair was either similar or dissimilar; similar pairs have the same video descriptions associated with them (*e.g.* brightness, speed of movement) while dissimilar videos have

Table 2. Comparisons of number of new lines of code written, processing times and accuracies of solutions between single-executable image processing program and multiple-executable workflow system (SWAV) to adapt to changing domain descriptions.

Domain Descriptions (User Preference)	Image Processing Expert			Workflow Modeller		
	New Lines of Code	Time (min.)	Accuracy %	New Lines of Code	Time (min.)	Accuracy %
Prefer false alarm than miss	43	16	58.25	3	3	59.30
Prefer miss than false alarm	56	23	62.55	2	2	64.80
Clear, no background movement	43	16	58.46	3	3	60.71
Clear, background movement	61	27	60.42	2	2	60.10
Blur, no background movement	43	16	60.88	3	3	62.09
Blur, background movement	57	32	63.80	2	2	61.22
Average	50.50	21.67	60.73	2.50	2.50	61.37

different descriptions. Similar videos will require the same detection algorithm for the most optimal result while dissimilar pairs do not require the same detection algorithms for optimal result. The aim was to test if subjects were able to determine the most optimal tool for the detection algorithm based on the recommended descriptions provided by SWAV. If they were, then they should select the same tool as the most optimal one if the second video is similar. They should also not conclude to select the same tool as the most optimal one if the second video is dissimilar. At each run, the workflow tool will display the video to the subject before proceeding to solve the task that would enable them to recognise the video descriptions. When the user has to select the detection algorithm, the workflow tool provides a set of recommended descriptions for each exiting tool (via the semi-automatic planning mode). Using these recommendations and their knowledge of the video descriptions, the user should be able to make an informed decision. Each subject on average selected the correct optimal tool for the second video 5 out of 7 times within the similar videos, and only 2.25 times out of 7 times within the dissimilar videos. Statistical testing using the t -distribution has yielded a significance level of $p = 0.0004$, proving that the workflow tool has helped the user learn and manage the processes involved in selecting the optimal steps when solving a VIP task.

8 Conclusions

This paper has outlined SWAV, an efficient, adaptable and user sensitive workflow system for solving VIP tasks. Its novelties include (semi-)automatic workflow composition, new flexible way of solving VIP tasks and enabling naive users to learn optimal VIP solutions. Efforts to incorporate SWAV onto distributed infrastructures such as the Cloud for processing large-scale videos is underway.

References

1. Ecogrid: National Center for High Performance Computing, Taiwan (2006), <http://ecogrid.nchc.org.tw>
2. Howell, D.C.: *Statistical Methods for Psychology*, 6th edn., Belmont, CA (2007)
3. Nadarajan, G., Chen-Burger, Y.H., Fisher, R.B.: A Knowledge-Based Planner for Processing Unconstrained Underwater Videos. In: *STRUCK 2009* (2009)
4. Spampinato, C., Chen-Burger, Y.H., Nadarajan, G., Fisher, R.B.: Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. In: *VIS-APP 2008*, pp. 514–519 (2008)
5. Taylor, I., Deelman, E., Gannon, D., Shields, M.: *Workflows for e-Science*. Springer, New York (2007)

A Meta-Model of Business Interaction for Assisting Intelligent Workflow Systems

Areti Manataki and Yun-Heh Chen-Burger

Centre for Intelligent Systems and their Applications, School of Informatics,
The University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK
A.Manataki@sms.ed.ac.uk, Jessicac@inf.ed.ac.uk

Abstract. Business interaction is a significant aspect of intelligent workflow systems, which are envisaged to deal with dynamic and interorganisational environments. However, automating business interaction in an intelligent way is not an easy task. To address this issue, this paper presents an ontological approach for modelling business interaction. The developed meta-model can support the process of automating business interaction, as well as assist its automated analysis. The ontology's development is described and its application is demonstrated through an example use case.

Keywords: intelligent workflow, business interaction, ontology, business action theory.

1 Introduction

Today's business environment is characterised as dynamic, uncertain, distributed, heterogeneous and collaborative. This business reality, often taking the form of virtual organisations and extended enterprises, enforces challenging constraints on workflow management systems. On this basis, a growing body of research has attempted to provide intelligent workflow systems addressing the following properties [1, 2, 3, 4]: Firstly, distributed, interorganisational and decentralised architectures are preferred, such as multiagent systems. Secondly, well-defined semantics are needed to support a common understanding of domain knowledge. Thirdly, advanced reasoning and learning capabilities are employed to allow for adaptive and flexible workflows. Lastly, a high degree of automation is required for the different phases of the workflows' lifecycle.

Business interaction is an important aspect of intelligent business workflows, especially given the highly distributed nature of the business environment. In a multiagent setting, for example, an ineffective communication and interaction between the agents can lead to a failure of the overall workflow. However, automating business interaction in an intelligent way is not an easy task. Such an intelligence requires the above-mentioned properties for the automated business interaction, so as to allow its analysis and automated improvement.

In this paper we argue that a formal model of business interaction can support the process of automating business interaction. Recognising Business Action Theory

(BAT) as a useful framework of business interaction, we formalise it in the form of an ontology. This meta-model has clear semantics, suitable for automated reasoning, and hence can facilitate the analysis of business interaction.

The remainder of this paper is as follows: Section 2 provides background information on the adopted modelling framework, and describes Business Action Theory, upon which the ontology is based. Section 3 presents the development of the BAT ontology, and discusses outputs of different phases. A scenario of use is described in Section 4, while Section 5 concludes.

2 Background Information

Business Action Theory [5, 6, 7, 8] is a generic framework of business interaction. Based on communicative action theories and business relationship theories, it describes the interaction between a supplier and a customer with respect to communication and exchange of value. It consists of a phase model and supplementary conceptual models. Six phases of business interaction are identified: i) business prerequisites phase, ii) exposure and contact search phase, iii) contact establishment and proposal phase, iv) contractual phase, v) fulfilment phase and vi) completion phase. Each phase involves some exchange, respectively: i) knowledge about business prerequisites, ii) interests, iii) proposals, iv) commitments, v) value and vi) assessments, which can be acceptances or claims. In [6] three additional conceptual models are provided, describing i) the transition from business prerequisites to a contract, ii) the contract as a base for fulfilment and iii) the results of fulfilment. According to the first conceptual model, the supplier's ability is translated into an offer, which is equivalent to a sales proposal. From the customer's side, an operation can have some lack and need, giving rise to desire and demand. This desire and demand can be expressed as a purchase proposal. Since several proposals may be exchanged, it is typical that a received proposal may lead to a reformulation of the supplier's offers and the customer's desire and demand. When a proposal is accepted, it is transformed into a contract, while in the opposite case no deal is achieved.

Furthermore, BAT recognises five layers of business interaction [8]: i) business acts, ii) action pairs, consisting of two interrelated business acts, iii) exchanges, consisting of one or more action pairs of the same type, iv) business transactions, consisting of exchanges of interests (possibly), proposals (possibly), commitments, value and assessments, and v) transaction groups, consisting of one or more business transactions and frame contracting. Being a theoretical framework of business interaction, BAT is useful for understanding and designing business interactions; however, in its current form, it cannot be used as a direct input when automating and implementing business interactions within a workflow system.

The *three-layered Business Process Modelling framework* [9] supports the development of workflow management systems, as well as generic software systems, by separating the business logic from the implementation logic. Its three layers are: i) The business layer captures the business requirements in higher-level descriptions that may be formal or informal. ii) The logical layer formalises the captured business

requirements in a machine-understandable language, while incorporating logical or operational requirements. iii) The implementation layer provides algorithms for the implementation of the logical model, while respecting overall system requirements.

3 Business Action Theory Ontology

With the aim of automating business interaction for intelligent workflows, and following the three-layered Business Process Modelling framework, we propose and develop a meta-model of business interaction. Business Action Theory is identified, in its current form, at the business layer of the framework. In order to transform it into a model at the logical layer, we develop a BAT ontology. This logical model can be particularly useful at web, knowledge-intensive environments.

The METHONTOLOGY [10, 11] approach is adopted for developing the BAT ontology, as it is a detailed, mature and widely accepted methodology for ontological engineering. METHONTOLOGY identifies five phases during the ontology life cycle (i.e. specification, conceptualisation, formalisation, implementation and maintenance), during which specific tasks take place. In the rest of this section, we will discuss outputs of the first four phases.

3.1 Specification and Knowledge Acquisition

The purpose of the BAT ontology is to support the development of intelligent workflows. The primary objective is to provide a formal, logical model of business interaction that can facilitate the automation of business interaction, while a secondary objective is the automated analysis of business interaction.

The source of knowledge for the BAT ontology is the literature on Business Action Theory, as presented in Section 2. It is worth mentioning that BAT has been refined and enriched since it was first introduced in [5]. However, this refinement has not been exhaustive, and several of the concepts and models that can be seen in the first versions of BAT are still applicable in the latest versions. Therefore, [5, 6, 7, 8] are the sources of knowledge for this ontology.

As far as the scope of the ontology is concerned, we distinguish the following four concept categories: i) phase model, as presented in [6], ii) additional concepts of the phase model, as presented in [7], iii) supplementary conceptual models, as presented in [6] and iv) business interaction layers, as presented in [8]. Finally, the granularity level of the ontology is the same as the one of BAT literature.

3.2 Conceptualisation

The acquired knowledge is conceptualised during this phase into a set of intermediate representations of the domain. METHONTOLOGY suggests a sequence of eleven tasks to support this activity, with outputs including among others a glossary of terms, a concept dictionary and a list of axioms. In this section the outputs of the first four tasks will be discussed.

Terms of the BAT domain were firstly gathered in a *Glossary of Terms* that contained 132 concepts, attributes and relations. An extract can be seen in Table 1, where some basic terms are described. For example, a description of the concepts “Business Role” and “Business Interaction” is provided, along with the relation between them “participatesInInteraction”, i.e. a Business Role participates in some Business Interaction.

Table 1. Extract from the Glossary of Terms

Name	Synonym	Acronym	Description	Type
Business Interaction	--	--	An interaction between a supplier and a customer for a single business transaction	Concept
Business Interaction Phase	--	--	A phase of the business interaction process	Concept
Contractual Phase	--	--	A phase where a contract is signed among the business parties (4th phase of BAT’s phase model)	Concept
Commitment Exchange	--	--	An exchange of commitments	Concept
Commitment	Promise, obligation	--	An engagement caused by contract	Concept
Business Role	--	--	The role that a business actor plays during a business transaction	Concept
participatesInInteraction (<i>BusinessRole, BusinessInteraction</i>)	--	--	An interaction in which a business role participates	Relation
hasPhase (<i>BusinessInteraction, BusinessInteractionPhase</i>)	--	--	A phase of a business interaction	Relation
isAccepted	--	--	Whether a proposal is accepted or not (boolean type)	Instance Attribute

A *Concept Taxonomy* was then defined, an extract of which can be seen in Figure 1. According to this classification, the concepts “Payment Promise” and “Delivery Promise” make up a partition of the concept “Commitment”, which was described in Table 1. Furthermore, “Commitment” belongs to the partition of “Object of Exchange”.

Relationships between concepts were also established and visualised in a set of *Binary Relations Diagrams*. Two such diagrams are presented in Figures 2 and 3. Figure 2 shows, among others, the relation “participatesInInteraction” that was presented in Table 1, while Figure 2 corresponds to a conceptual model from [6] that was described in Section 2.

Based on the outputs of the previous tasks, a *Concept Dictionary* was developed for BAT. Table 2 presents some concepts, along with the relations in which they participate.

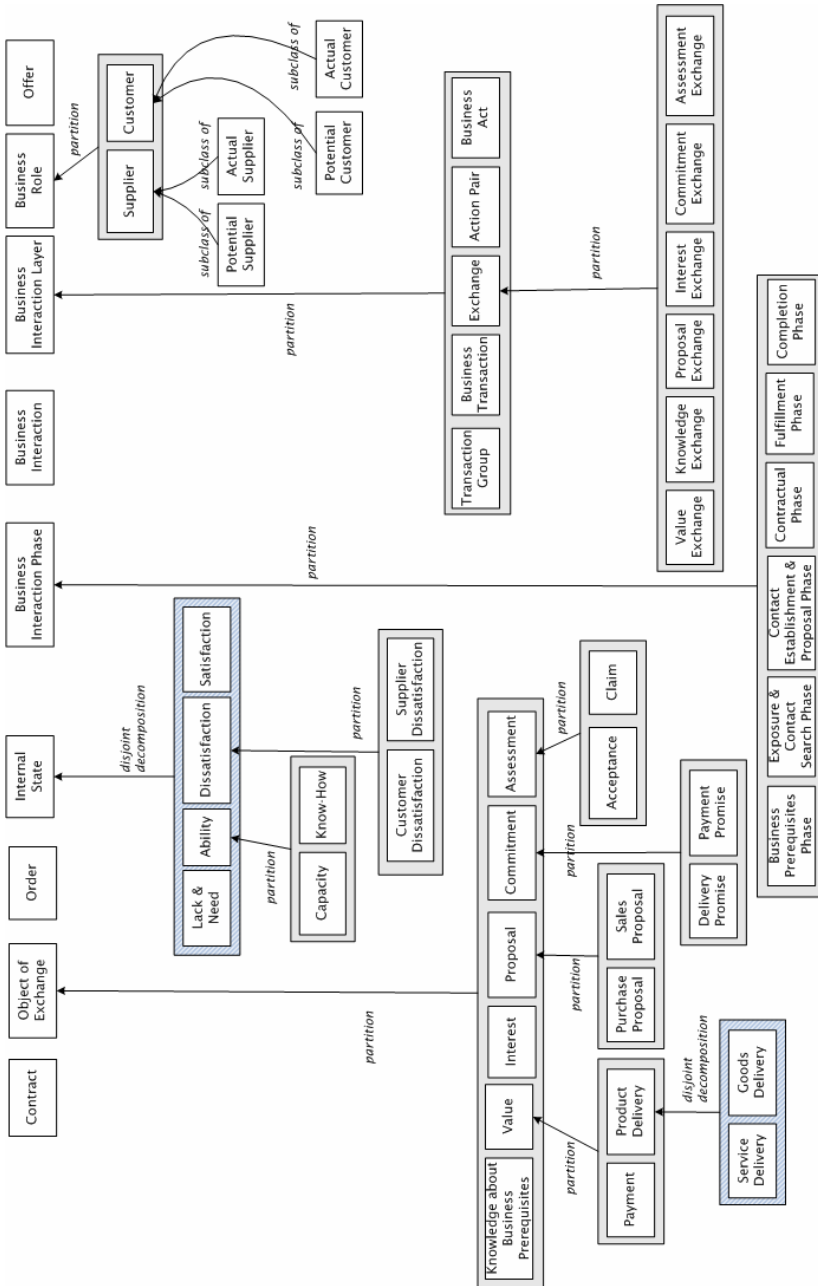


Fig. 1. Extract from the Concept Taxonomy

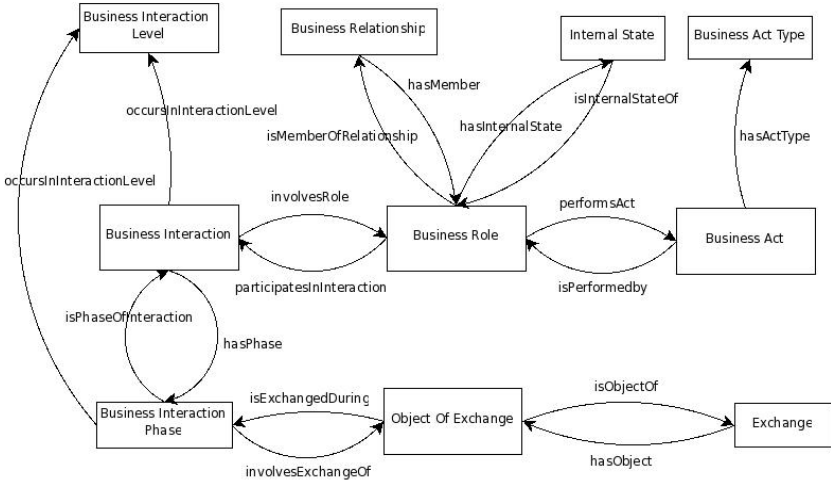


Fig. 2. Extract from the Binary Relations Diagram

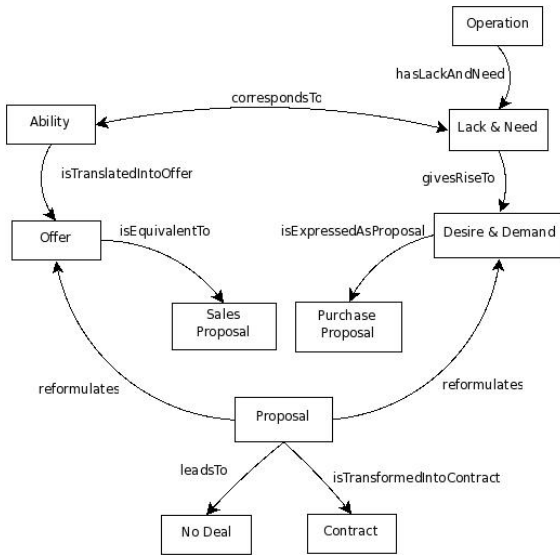


Fig. 3. Extract from the Binary Relations Diagram, describing BAT's model of transition from business prerequisites to a contract

Table 2. Extract from the Concept Dictionary

Concept Name	Relations
Business Role	participatesInInteraction, hasInternalState, isMemberOfRelationship, performsAct, hasInternalState, hasOperation
Contractual Phase	isPhaseOfInteraction, involvesExchangeOf, occursInInteractionLevel
Contract	expressesCommitment, isRealizedIn, putsNewRequirementsOnAbility
Ability	isInternalStateOf, isTranslatedIntoOffer, correspondsTo, isUsedFor

3.3 Formalisation

The formalisation phase involves the translation of the conceptual model into a formal or semi-computable model. This phase is important for two reasons: firstly, conceptual models are often ambiguous; secondly, an explicit domain model with well-understood formal semantics can be useful during the implementation phase. Description Logics (DL) [12] are generally regarded as an appropriate formalism for this task, covering both terminological knowledge (i.e. concept descriptions) and assertional knowledge (i.e. specific knowledge about individuals) about a domain. They also allow for efficient reasoning with respect to concept satisfiability, subsumption, equivalence and disjointness, as well as consistency and instance checking of the assertional knowledge. This reasoning support can be of great value, as modelling errors can be detected and resolved.

For the above reasons, the developed BAT conceptual model was transformed into a formal model expressed in the DL language *ALCN*. An extract of this formal model is provided in the following code, where terminological axioms can be seen. For example, according to the first statement, the concept “PaymentPromise” is subsumed by (i.e. subclass of) the concept “Commitment”, and according to the second, the concepts “PaymentPromise” and “DeliveryPromise” are disjoint. The fourth statement is a necessary and sufficient definition of the concept “BusinessTransaction”, according to which a BusinessTransaction is defined as a “BusinessInteractionLayer” that consists of at most 5 “Exchanges” and that consists of some “CommitmentExchange”, some “ValueExchange” and some “AssessmentExchange”.

```

PaymentPromise  $\sqsubseteq$  Commitment
Paymentpromise  $\sqcap$  DeliveryPromise  $\sqsubseteq \perp$ 
CommitmentExchange  $\equiv$  Exchange  $\sqcap \exists$ hasObject.Commitment
BusinessTransaction  $\equiv$  BusinessInteractionLayer  $\sqcap$ 
 $\leq 5$ consistsOfExchange  $\sqcap$ 
 $\exists$ consistsOfExchange.CommitmentExchange  $\sqcap$ 
 $\exists$ consistsOfExchange.ValueExchange  $\sqcap$ 
 $\exists$ consistsOfExchange.AssessmentExchange

```

3.4 Implementation

Utilising the conceptual and formal models discussed in Sections 3.2 and 3.3, the BAT ontology was implemented during this phase. The language chosen was OWL [13], as it has become a standard of the Semantic Web community and also because of its correspondence with DL. The ontology editor Protégé [14] was used for this task, a screenshot of which can be seen in Figure 4. Some sample code is provided below, specifying the class “PaymentPromise”, and capturing the first two DL statements of Section 3.3.

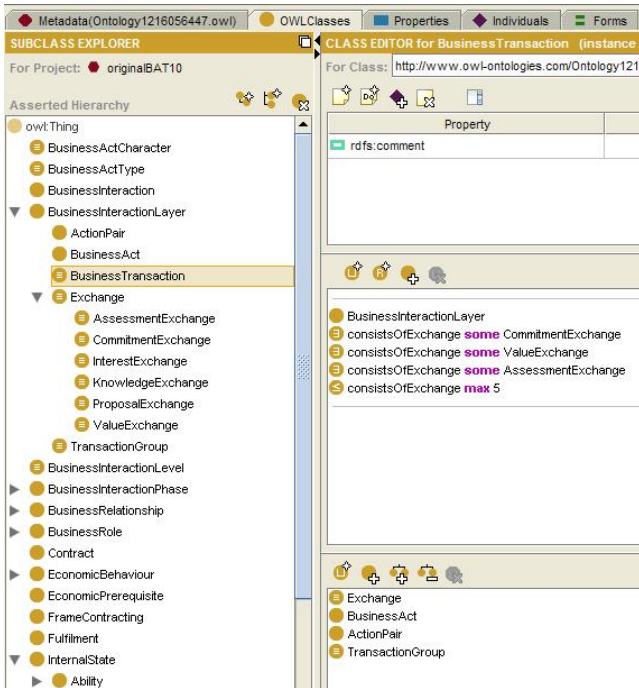


Fig. 4. Screenshot from Protégé

```
<owl:Class rdf:about="#PaymentPromise">
  <rdfs:subClassOf rdf:resource="#Commitment" />
  <owl:disjointWith rdf:resource="#DeliveryPromise" />
</owl:Class>
```

4 Example Use Case

As specified in Section 3.1, the BAT ontology is intended to be used firstly for facilitating the automation of business interaction, and secondly for supporting the automated analysis and improvement of business interaction. A scenario of use of the second type follows.

Imagine an agent-based virtual organisation (VO) within the travel agency sector. The VO comprises of several independent member companies, each one represented by an intelligent agent. During the VO's life cycle there is a high level of interaction between the agents; some interactions involve long-lasting relationships between VO members, while others involve the choice of new partners for collaboration. As part of their intelligent behaviour, VO members/agents analyse their business interactions with respect to the BAT ontology in order to assess and improve their performance.

Let's suppose now that agent1 analyses his business interactions at different layers (as defined at the BAT ontology), and identifies the following issue at the business transaction layer with agent2: Even though at the completion phase there is a high level of satisfaction from agent2's delivered service, the overall transaction has long duration. By following the definitions of business interaction layers from the BAT ontology, agent1 shifts the analysis to the next layer (i.e. the Exchange layer), and finds that the long duration is due to the long exchange of proposals, which corresponds to the contact establishment and proposal phase. Having identified the source of the problem, agent1 can decide on a future change of his negotiation strategy during this phase, and he can also let agent2 know about this problem. A message to agent2 suggesting to shorten the proposal phase of their future transactions could, thus, lead to a collaborative solution to the arisen problem.

But how can we assure that agent2 will understand the content of that message? If the message references the BAT ontology for the relevant used terms, agent2 will be able to look up their definition and understand their meaning, thus leading to an effective communication between the two agents. To sum up, two different uses of the BAT ontology were identified in this scenario. First, as a reference point for analysing an agent's business interactions, and second as a common vocabulary between agents to support their communication.

5 Conclusions

Recognising the significance of business interaction for intelligent business workflows, we have suggested in this paper a meta-model of business interaction. We have developed this meta-model in the form of an ontology based on appropriate business theories, and we have presented in this paper outputs throughout its life cycle. Its value is dual: Firstly, it can support the implementation of automated business interaction, and secondly it can assist the automated analysis of business interaction, thus injecting intelligence into workflows. Its well-defined semantics make it suitable for web-based, distributed and knowledge-intensive environments, such as agent-based workflows. A scenario of use in such an environment has also been presented in this paper. In the future we would like to explore the use of the BAT ontology within a supply chain simulation setting, and study issues of supply chain members' power and dominance, as determined by their business interactions. Evaluating the meta-model's completeness and coverage within such a setting, and comparing it to those of alternative approaches, such as ebXML, is another possible future avenue of research.

References

1. van Der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) *BPM 2003*. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Chung, P.W.H., Cheung, L., Stader, J., Jarvis, P., Moore, J., Macintosh, A.: Knowledge-based process management—an approach to handling adaptive workflow. *Knowledge-Based Systems* 16(3), 149–160 (2003)
3. Wang, M., Wang, H., Xu, D.: The design of intelligent workflow monitoring with agent technology. *Knowledge-Based Systems* 18(6), 257–266 (2005)
4. Cardoso, J., Sheth, A.: Semantic E-Workflow Composition. *Journal of Intelligent Information Systems* 21(3), 191–225 (2003)
5. Goldkuhl, G.: Generic business frameworks and action modelling. In: Dignum, F., Dietz, J., Verharen, E., Weigand, H. (eds.) *Communication Modeling – The Language/Action Perspective*, Proceedings of the First International Workshop on Communication Modeling, Electronic Workshops in Computing, Springer, Berlin (1996)
6. Goldkuhl, G.: The six phases of business processes – business communication and the exchange of value. In: 12th Biennial ITS Conference “Beyond Convergence” (ITS 1998), Stockholm (1998)
7. Goldkuhl, G., Lind, M.: Developing e-interactions – a framework for business capabilities and exchanges. In: Proceedings of the 12th European Conference on Information Systems (ECIS 2004), Turku (2004)
8. Lind, M., Goldkuhl, G.: The constituents of business interaction—generic layered patterns. *Data & Knowledge Engineering* 47(3), 327–348 (2003)
9. Chen-Burger, Y.-H., Stader, J.: Formal Support for Adaptive Workflow Systems in a Distributed Environment. In: Fischer, L. (ed.) *Workflow Handbook 2003*, pp. 93–118. Future Strategies Inc., Florida (2003)
10. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Spring Symposium on Ontological Engineering of AAAI, pp. 33–40. Stanford University, California (1997)
11. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer, Heidelberg (2004)
12. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: *The Description Logic Handbook*. Cambridge University Press, New York (2003)
13. OWL Web Ontology Language Overview,
<http://www.w3.org/TR/owl-features/>
14. Protégé, <http://protege.stanford.edu/>

Author Index

- Abdulrahman, Ruqayya 54, 84
Abe, Tsukasa 211
Almiron, Santiago 103
Amos, Martyn 514
Antosiewicz, Marek 64
Aoyama, Hiroki 583
Asakura, Koichi 583
Ayesh, Aladdin 504
- Bąk, Jarosław 301
Bañares, José A. 661
Bandar, Zuhair 16, 26, 35, 44, 534, 544
Barbati, Maria 486
Barbucha, Dariusz 476
Beheshti, R. 311
Biernacki, Paweł 406
Boella, Guido 280
Bojic, Iva 159
Bouslimi, Issam 148
Brahimi, Said 113
Bruno, Giuseppe 486
- Čapkovič, František 221
Carbo, Javier 373
Čavrak, Igor 169
Chen, Pei-Yi 640
Chen, Shiuann-Shuoh 640
Chen, Weiru 291
Chen-Burger, Yun-Heh 681, 692
Chizhov, Yuri 446
Choinski, Dariusz 231, 353
Chuang, Yu-Wei 640
Crispin, Alan 553
Crockett, Keeley 16, 26, 35, 44, 414, 544, 563
Crossley, Matthew 514
Cybulka, Jolanta 301
Czarnowski, Ireneusz 436
Czubak, Adam 425
- Dickinson, Patrick 524
Duckett, Tom 524
- Edmonds, Bruce 563
Eguchi, Tomoya 573
- Fernández-Caballero, Antonio 200
Fisher, Robert B. 681
Fuks, Konrad 602
- Gabbay, Dov M. 280
Gascueña, José M. 200
Genovese, Andrea 486
Ghédira, Khaled 148
Golińska, Paulina 620
Griol, David 373
Guan, Hui 291
- Hajdul, Marcin 620
Hanachi, Chihab 148
Hara, Noriyuki 261
Hayashida, Tomohiro 261
Hijjawi, Mohammad 35
Hinde, Chris 363
Holton, D.R.W. 54, 84
Horák, Aleš 130
Hsieh, Fu-Shiung 383
- Jędrzejek, Czesław 301
Jędrzejowicz, Joanna 495
Jędrzejowicz, Piotr 2, 436, 456, 466, 495
Jezic, Gordan 159, 332
Jin, Zhi 394
- Kambayashi, Yasushi 211
Kaminski, Bogumil 64
Kawa, Arkadiusz 602
Kazienko, Przemysław 651
Keane, John 414
Kiv, Sodany 321
Kolp, Manuel 321
Kułakowski, Konrad 190
Kusek, Mario 159
- Laroum, Toufik 94
Lasota, Tadeusz 342
Latham, Annabel 563
Lijiao, Chang 592
Lipic, Tomislav 159
Liu, Lin 291, 394
Lovrek, Ignac 74

- Lu, Ruqian 394
 Luczak, Tomasz 342
- Maamri, Ramdane 113, 123
 Małachowski, Bartłomiej 630
 Manataki, Areti 692
 Maries, Iulia 271
 McLean, David 35, 534, 563
 Metzger, Mieczysław 241, 353
 Mianowska, Bernadetta 138
 Michalski, Radosław 651
 Molina, Jose M. 373
 Mozayani, N. 311
 Muyebe, Maybin 414
- Nadarajan, Gayathri 681
 Navarro, Elena 200
 Neagu, Daniel 54, 84
 Nguyen, Ngoc Thanh 138
 Nikolaev, Pavel 504
 Nishizaki, Ichiro 261
 Nocon, Witold 353
- Osathanunkul, Khukrit 544
 O'Shea, James 16, 26, 544
 Owda, Majdi 44
- Palus, Sebastian 651
 Pearce, David M. 534
 Perotti, Alan 280
 Petric, Ana 332
 Piech, Henryk 251, 425
 Podobnik, Vedran 74
 Polaków, Grzegorz 241
 Prýmek, Miroslav 130
- Rana, Omer F. 661
 Ratajczak-Ropel, Ewa 466
 Renaud, Dominique 103
 Resconi, Germano 363
- Ridley, Mick 84
 Różewski, Przemysław 630
- Sahnoun, Zaidi 113
 Sanchez-Pi, Nayat 373
 Scarlet, Emil 271
 Senik, Michał 231
 Sikora, Tadeusz 130
 Stępień, Tomasz 190
 Stranjak, Armin 169
 Szupiluk, Ryszard 610
 Szymanezyk, Olivier 524
- Takahashi, Hiroshi 180
 Takahashi, Satoru 180
 Takimoto, Munehiro 211
 Telghamti, Samira 123
 Terano, Takao 180
 Thabet, Inès 148
 Tighiouart, Bornia 94
 Titiloye, Olawale 553
 Tolosana-Calasanz, Rafael 661
 Trawiński, Bogdan 342
- Ushiyama, Taketoshi 573
- van der Torre, Leendert 280
 Villata, Serena 280
- Watanabe, Toyohide 583, 592
 Wautelet, Yves 321
 Wierzbowska, Izabela 456
 Wojewnik, Piotr 64, 610
 Wooldridge, Michael 1
- Yang, Hongji 291, 394
 Yeh, Ching-Long 671
- Ząbkowski, Tomasz 610
 Żagar, Mario 169
 Zanni-Merk, Cecilia 103
 Zawisza, Mateusz 64