Studies in Computational Intelligence 329

Santi Caballé
Fatos Xhafa
Ajith Abraham (Eds.)

Intelligent Networking, Collaborative Systems and Applications

Springer

Santi Caballé, Fatos Xhafa, and Ajith Abraham (Eds.)

Intelligent Networking, Collaborative Systems and Applications

Studies in Computational Intelligence, Volume 329

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 307. Anupam Shukla, Ritu Tiwari, and Rahul Kala
Towards Hybrid and Adaptive Computing, 2010
ISBN 978-3-642-14343-4

Vol. 308. Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi (Eds.)
Advances in Intelligent Tutoring Systems, 2010
ISBN 978-3-642-14362-5

Vol. 309. Isabelle Bichindaritz, Lakhmi C. Jain, Sachin Vaidya, and Ashlesha Jain (Eds.)
Computational Intelligence in Healthcare 4, 2010
ISBN 978-3-642-14463-9

Vol. 310. Dipti Srinivasan and Lakhmi C. Jain (Eds.)
Innovations in Multi-Agent Systems and Applications – 1, 2010
ISBN 978-3-642-14434-9

Vol. 311. Juan D. Velásquez and Lakhmi C. Jain (Eds.)
Advanced Techniques in Web Intelligence, 2010
ISBN 978-3-642-14460-8

Vol. 312. Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Recognition based on Biometrics, 2010
ISBN 978-3-642-15110-1

Vol. 313. Imre J. Rudas, János Fodor, and Janusz Kacprzyk (Eds.)
Computational Intelligence in Engineering, 2010
ISBN 978-3-642-15219-1

Vol. 314. Lorenzo Magnani, Walter Carnielli, and Claudio Pizzi (Eds.)
Model-Based Reasoning in Science and Technology, 2010
ISBN 978-3-642-15222-1

Vol. 315. Mohammad Essaaidi, Michele Malgeri, and Costin Badica (Eds.)
Intelligent Distributed Computing IV, 2010
ISBN 978-3-642-15210-8

Vol. 316. Philipp Wolfrum
Information Routing, Correspondence Finding, and Object Recognition in the Brain, 2010
ISBN 978-3-642-15253-5

Vol. 317. Roger Lee (Ed.)
Computer and Information Science 2010
ISBN 978-3-642-15404-1

Vol. 318. Oscar Castillo, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Intelligent Control and Mobile Robotics, 2010
ISBN 978-3-642-15533-8

Vol. 319. Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, Tokuro Matsuo, and Hirofumi Yamaki (Eds.)
Innovations in Agent-Based Complex Automated Negotiations, 2010
ISBN 978-3-642-15611-3

Vol. 320. xxx

Vol. 321. Dimitri Plemenos and Georgios Miaoulis (Eds.)
Intelligent Computer Graphics 2010
ISBN 978-3-642-15689-2

Vol. 322. Bruno Baruque and Emilio Corchado (Eds.)
Fusion Methods for Unsupervised Learning Ensembles, 2010
ISBN 978-3-642-16204-6

Vol. 323. Yingxu Wang, Du Zhang, and Witold Kinsner (Eds.)
Advances in Cognitive Informatics, 2010
ISBN 978-3-642-16082-0

Vol. 324. Alessandro Soro, Vargiu Eloisa, Giuliano Armano, and Gavino Paddeu (Eds.)
Information Retrieval and Mining in Distributed Environments, 2010
ISBN 978-3-642-16088-2

Vol. 325. Quan Bai and Naoki Fukuta (Eds.)
Advances in Practical Multi-Agent Systems, 2010
ISBN 978-3-642-16097-4

Vol. 326. Sheryl Brahnam and Lakhmi C. Jain (Eds.)
Advanced Computational Intelligence Paradigms in Healthcare 5, 2010
ISBN 978-3-642-16094-3

Vol. 327. Slawomir Wiak and Ewa Napieralska-Juszczak (Eds.)
Computational Methods for the Innovative Design of Electrical Devices, 2010
ISBN 978-3-642-16224-4

Vol. 328. Raoul Huys and Viktor K. Jirsa (Eds.)
Nonlinear Dynamics in Human Behavior, 2010
ISBN 978-3-642-16261-9

Vol. 329. Santi Caballé, Fatos Xhafa, and Ajith Abraham (Eds.)
Intelligent Networking, Collaborative Systems and Applications, 2010
ISBN 978-3-642-16792-8

Santi Caballé, Fatos Xhafa, and Ajith Abraham (Eds.)

Intelligent Networking, Collaborative Systems and Applications

Springer

Santi Caballé
Open University of Catalonia
Department of Computer Sciences
Multimedia and Telecommunications
Rbla. Poblenou. 156
08018 Barcelona
Spain
E-mail: scaballe@uoc.edu

Dr. Fatos Xhafa
Department of Languages and
Informatics Systems
Polytechnic University of Catalonia
Campus Nord, Ed. Omega,
C/Jordi Girona 1-3
08034 Barcelona
Spain
E-mail: fatos@lsi.upc.edu

Prof. Ajith Abraham
Machine Intelligence Research Labs
(MIR Labs)
Scientific Network for Innovation and
Research Excellence
P.O. Box 2259
Auburn, Washington 98071-2259
USA
E-mail: ajith.abraham@ieee.org

ISBN 978-3-642-16792-8 e-ISBN 978-3-642-16793-5

DOI 10.1007/978-3-642-16793-5

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2010938642

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Foreword

For the last fifty years the most, if not all, computers in the world have been built based on the von Neumann model, which in turn was inspired by theoretical model proposed by Alan Turing early in the twentieth century. A Turing machine is the most famous theoretical model of computation that can be used to study a wide range of algorithmic styles.

The von Neumann model has been used to build computers with great success. It has also been extended to the development of the early supercomputers and we can also see its influence on the design of some of the high performance computers of today. However, the principles espoused by the von Neumann model are not adequate for solving many of the problems that have great theoretical and practical importance. In general, a von Neumann model is required to execute a precise algorithm that can manipulate accurate data. In many problems such conditions cannot be met. For example, in many cases accurate data is not available or a "fixed" or "static" algorithm cannot capture the complexity of the problem under study.

The current volume *Intelligent Networking, Collaborative Systems and Applications* is an excellent demonstration of how far we have come from the days of von Neumann. The richness and variety of problem domains and solution methodologies that this volume demonstrates are truly remarkable. Most importantly, this volume shows how the process of computation today has changed to become more *distributed* in nature. For example, this volume explores this by investigating in thorough fashion a number of important topics:

- Intelligent networking for virtual organizations and campuses
- Intelligent networking for social networks
- Intelligent collaborative systems for work and learning
- Security, organization, management and autonomic computing for intelligent networking and collaborative systems
- Grid and P2P distributed infrastructure for intelligent networking and collaborative systems
- Wireless and sensor systems for intelligent networking and collaborative systems
- Data mining and knowledge management for intelligent networking and collaborative systems

One can notice that in the above topics that apart from the distributed nature of computation there is a great requirement that the different components of a distributed system need to *collaborate* and work in unison in order for the whole system to function properly. Another keyword that is quite prominent in this volume is that of *intelligence*. As the complexity of today's systems increases the need to inject more intelligence in their design and operation becomes quite essential for the long term health of these systems. Some of the popular approaches in making complex systems more intelligent are through the use of nature- and bio-inspired techniques. Some of these techniques are now commonplace and have been accepted by the wider scientific community, such as artificial life techniques. These techniques are widely used to solve a variety of optimization problems, where they tend to excel in situations when the knowledge space is ambiguous or incomplete which is quite common in many real-world applications.

The editors of this volume bring together a number of researchers who explore the new developments and future directions of their respective fields. The list of topics that is explored here is by no means exhaustive but most of the conclusions provided here could be extended to the other research fields that are not covered here. I also hope that the topics covered will get the readers to think of the implications of such new ideas on the developments in their respective fields. The publication of *Intelligent Networking, Collaborative Systems and Applications* will be an important addition to the literature on this emerging topic.

Albert Y. Zomaya
The University of Sydney, Australia

Preface

Introduction

With the fast development of the Internet, we are experiencing a shift from the traditional sharing of information and applications as the main purpose of the Web to an emergent paradigm, which locates people at the very center of networks and exploits the value of people's connections and relations. Web 2.0 has come to play a major role in this context by enabling a new generation of social networks and web-based communities and dramatically changing the way we use and interact with the Web. Social network analysis is also a rapidly growing field to investigate the dynamics and structure of intelligent Web-based networking and collaborative systems.

Virtual campuses and organizations strongly leverage intelligent networking and collaborative systems by a great variety of formal and informal electronic relations, such as business-to-business, peer-to-peer and many types of online collaborative learning interactions. This has resulted in entangled systems that need to be managed efficiently and in an autonomous way. In addition, latest and powerful technologies based on Grid and Wireless infrastructure as well as Cloud computing are currently enhancing collaborative and networking applications a great deal but also facing new issues and challenges. Well-known social networks lack of knowledge management and adaptive solutions and the information shared among peers is rather static. Virtual communities of practice also provide poorly interactive solutions and lack of full support for organization, management, mobility and security.

The ultimate aim of this book is to stimulate research that leads to the creation of responsive environments for networking and, at longer-term, the development of adaptive, secure, mobile, and intuitive intelligent systems for collaborative work and learning.

This book follows the International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), held on November 4-6, 2009 at the Open University of Catalonia, Barcelona, Spain (http://www.uoc.edu/symposia/incos2009).

Main Contributions of This Book

This book covers the latest advances in intelligent networking and collaborative systems that lead to gain competitive advantages in business and academia scenarios.

Overall, the book contributes with the following:

- *Social network* analysis is a rapidly growing field to investigate the dynamics and structure of intelligent Web-based networking and collaborative systems. The book provides knowledge management and adaptive solutions to enhance the information shared among peers in well-known social networks.
- *Virtual organizations* strongly leverage intelligent networking and collaborative systems by a great variety of formal and informal electronic relations, such as business-to-business, peer-to-peer and many types of online collaborative interactions. The book proposes solutions for the resulting entangled systems that need to be managed efficiently and in an autonomous way.
- *Emerging powerful infrastructures based on Grid, Cloud and Wireless technologies* are currently enhancing collaborative and networking applications a great deal but also facing new issues and challenges, some of them are addressed in the book. Also, the book provides solutions to virtual communities of practice that lack of full support for organization, management, mobility and security.
- *Security and resilience* are important and critical issues in networking and collaborative systems that the book covers extensively.

Organization of the Book

The 13 chapters of this book are organized as follows:

I. Intelligent Collaborative Systems and Applications

Chapter 1: Wikipedia as a Source of Ontological Knowledge: State of the Art and Application

This chapter motivates that Wikipedia can be used as a source of knowledge for creating semantic enabled applications, and consists of two parts. First, we provide an overview over different research fields which attempt to extract knowledge encoded by humans inside Wikipedia. The extracted knowledge can then be used for creating a new generation of intelligent applications based on the collaborative character of Wikipedia, rather than on domain ontologies which require the intervention of knowledge engineers and domain experts. Second, as a proof of concept, we describe an application whose intelligent behavior is achieved by using Wikipedia knowledge for automatic annotation and representation of multimedia presentations.

Chapter 2: Toward a Methodology of Collaborative Modeling and Simulation of Complex Systems

Complex Systems Modeling and simulation are activities where collaboration between researchers with complementary points of view, skills and knowledge is well established and constitutes the usual working way rather than an exception. To manage difficulties of traditional modeling projects (how to translate informal knowledge into implemented software, which intermediate model needed to be created...), lots of methodologies have been proposed. But to be well adapted to the particular case of complex systems, they need to go further. In this chapter, the aim is to provide a complete methodology taking into account the collaborative aspect of the Complex Systems Modeling and Simulation task. Then, collaboration between various participants is described and a minimal set of tools necessary for a modeling platform is proposed.

Chapter 3: Role-Based Collaboration Extended to Pervasive Computing

Computer Supported Collaborative Work (CSCW) has been a hot point in researches for some decades. Recent progresses in software and hardware technologies have allowed the use of more and more advanced applications and services. It also brought computing capabilities to mobile devices such as smartphones and laptops. This has led to an extensive use of computers to collaborate in some un-expected manners. Among the abundance of models designed to support collaboration some are particularly promising: tasks models, roles models and collaboration's context models. Simultaneously the Pervasive Computing paradigm has emerged from recent researches. In this chapter, a model to integrate the pervasive computing perspective into the collaborative work is proposed by the use of an original model: the PCSCW model (Pervasive Computing Supported Collaborative Work). This model relies on some robust concepts: a role model inspired by some recent works, a classical task model coupled to a precise resource model and the development of device collaboration rules. The resulting model provides a seamless and transparent cooperation of devices to simplify and facilitate the collaboration of humans.

II. Intelligent Networking and Resource Management

Chapter 4: Evolving Schedules of Independent Tasks by Differential Evolution

Scheduling is one of the core steps to efficiently exploit the capabilities of heterogeneous distributed computing systems and it is also an appealing NP-complete problem. There is a number of heuristic and meta-heuristic algorithms that were tailored to deal with scheduling of independent jobs. In this chapter, the authors investigate the efficiency of differential evolution for the scheduling problem and compare it with existing approaches. The analysis shows that the differential evolution is a promising method that can compete with well-established scheduling algorithms.

Chapter 5: A Lightweight Approach to Distributed Network Diagnosis under Uncertainty

Management applications have not kept the fast changing pace of networks and services and still rely on centralized and deterministic approaches. Besides, distribution and uncertainty are intrinsic issues in the telecommunications environment. Therefore, new approaches to network and service management have to be explored that can deal with these challenges. In this chapter, a lightweight collaborative framework for network trouble-shooting is presented. This framework is based on multi-agent platforms and probabilistic techniques and it has been prototyped and applied to three different network environments. A summary of the most relevant results obtained and conclusions reached is also provided.

Chapter 6: A Multi-lane Double Auction for Economic-Based Service Management in the Cloud

Economic models have shown their suitability to allocate resources efficiently, considering an unbalanced supply and demand. As the use of the Cloud is extending, a numerous set of distributed resource allocation frameworks have been developed to attain efficient resource management while keeping the scalability of the infrastructure. However, those frameworks make use of either simple double auction mechanisms or complex approximations to the NP-complete problem of the combinatorial auction. The problem of those mechanisms is that of its generality, that is, they have not been specially designed for the trading of time-leased computational resources. In this chapter, the authors present a novel variant of the double auction that has been specially adapted to trade time-differentiated items as Cloud services can be considered. The chapter presents the data structures, algorithms and architecture of the economic mechanism as well as it presents the evaluation of the mechanism through simulation. Simulated results are compared with the main double auction implementations found in the literature. The chapter constitutes an approach to improve efficiency of service management and allocation in the Cloud from the point of view of the economic model and not from architectural aspects addressed by most of the contributions found in the literature.

Chapter 7: Decentralized Self-optimization in Shared Resource Pools

Resource pools are collections of computational resources which can be shared by different applications. The goal is to accommodate the workload of each application, by splitting the total amount of resources in the pool among them. In this sense, utility functions have been pointed as the main tool for enabling self-optimizing behavior in such pools. The ultimate goal is to allow resources from the pool to be split among applications, in a way that the best outcome is obtained. Whereas different solutions in this context exist, it has been found that none of them tackles the problem we deal with in a total decentralized way. To this end, in this chapter, the authors present a decentralized and self-optimizing approach for resource management in shared resource pools.

Chapter 8: Monitoring and Performance Analysis of Workflow Applications in Large Scale Distributed Systems

The chapter presents the design, implementation and testing of the monitoring solution created for integration with a workflow execution platform. The monitoring solution is a key for modeling and performance analysis of Grid systems considered as a networking and collaborative systems. The monitoring solution constantly checks the system evolution in order to facilitate performance tuning and improvement. The novelty of the work presented in this chapter is the improvement of distributed application obtained using the real-time information to compute estimates of runtime which are used to improve scheduling. Monitoring is accomplished at application level, by monitoring each job from each workflow and at system level, by aggregating state information from each processing node. The scheduling performance in distributed systems can be improved through better runtime estimation and the error detection can automatically detect several types of errors.

Chapter 9: Scaling Relations of Data Gathering Times in an Epidemically Data Sharing System with Opportunistically Communicating Mobile Sensors

In this chapter, the authors investigate data gathering time in an epidemically data sharing system with opportunistically communicating mobile sensors. A stochastic process of the system is proposed where N sensors moved randomly and independently on the d-dimensional square grid with size L and when meeting opportunistically at the same position on the grid, the sensors shared and stored all possessing data epidemically. The authors focus on three data gathering times, that is, latency times that (1) at least one sensor collects all (2) every sensor collects at least one common data (3) every sensor collects all. As a result, in general, the complementary cumulative distribution functions of these times decay exponentially in their asymptotic regions. A decay speed is also examined, which is also called relaxation time, of the exponential decay numerically with varying d, L, and N. Finally, scaling relations of the relaxation times are shown. The authors conclude that these relations are useful for estimating the minimum required number of sensors to collect data within a certain short period of time when the sensors are densely covered on the system.

III. Intelligent Secure and Resilient Networking Systems

Chapter 10: Multilaterally Secure Ubiquitous Auditing

This chapter addresses tracking information of individuals as a useful input to many Ubiquitous Computing (UbiComp) applications. As an example, a smart emergency management application: once mobile first responders are continuously tracked, a precise and safe coordination of rescue missions is possible, and also mission logs can be created for audit purposes. However, continuously tracking users and storing the data for later use is often in conflict with individual privacy preferences. This may ultimately lead to the non-acceptance and rejection of these new technologies by their users. In order to reconcile privacy and accountability requirements in location tracking systems, the authors introduce and evaluate the approach of using auditing mechanisms on two levels. They illustrate that, by employing carefully designed cryptographic mechanisms for selective pseudonym linkability based on efficient techniques of secure multiparty computation, it is possible to balance the conflicting interests to a certain extent. The work reported in this chapter, motivated by and applied to smart emergency management systems, is a step towards the realization of multilaterally secure and thus multilaterally acceptable UbiComp systems supporting collaborative work.

Chapter 11: Intrusion Detection in Multi-Agent Systems

In this chapter, the authors present an adaptive intrusion detection system for distributed environments dedicated to developing agent-based applications. To this end, they propose a scalable, flexible and reactive agent based architecture and a lightweight genetic algorithm that recognizes the intruders in an adaptive and automatic way. The approach is based on monitoring the level of physical resources usage and implies the detection of those agents that manifest an abusive behavior. The authors finally enhance Jade with their intrusion detection system and the results obtained in different scenario cases are analyzed and illustrated.

Chapter 12: A Round-Based Cover Traffic Algorithm for Anonymity Systems

Anonymity is becoming more of an issue with the growing importance of networking. Examples include protecting privacy of Internet users or electronic voting. Several network anonymity systems have been deployed, the most popular of them is probably Tor. However, those systems do not protect against attackers who observe or modify the traffic to match sources with destinations. The protection method against such attacks by adding additional traffic is not usually implemented because of the high cost. In this chapter, the authors propose a new cover traffic generation algorithm for flow-based anonymity systems and compare it with other algorithms from the literature. The algorithm is based on four ideas: fixed time rounds, flow classification with different protection methods for different classes, protection depending on the potential cost and finally, use of history. In the chapter, the authors evaluate their algorithm both in theory and in practice. The analysis shows that this solution provides sufficient protection while reducing overhead traffic compared to the algorithms known from the literature.

Chapter 13: Fault Recovery Performance Analysis of Functionally Distributed Transport Networking System

This chapter proposes a fault recovery method in functionally distributed transport networking that separates the control-plane processing part (control element, CE) from the forwarding-plane processing part (forwarding element, FE) of the router. In this architecture, one path-control process in the CE consolidates and processes the path computations and the path settings for multiple FEs. This leads to reduction in the path-control complexity and efficient operation of large scale networks. On the other hand, it is absolutely critical to ensure the high reliability of the CE. The authors analyze the performance of the proposed fault recovery method by using software implementation.

Targeted Audience and Last Words

We expect that current complex virtual organizations and communities of practice strongly leverage the extensive research produced in this book, being the book's targeted audience, including industry and companies involved in intensive networking and collaborative systems. In particular, those organizations exploiting latest and powerful technologies based on Grid and Wireless infrastructures as well as Cloud computing can find many solutions in the book to alleviate complex issues and challenges arisen in this context, in terms of collaborative applications, resource management, mobility, security and system resilience.

Finally, academic researchers, professionals and practitioners in the field can also be inspired and put in practice the ideas and experiences proposed in the book in order to evaluate them for their specific research and work.

We hope that the readers find this book fruitful and help accomplish their goals. Enjoy the reading!

Acknowledgements

The editors of this book wish to thank the referees who have carefully reviewed the chapters and gave useful suggestions and feedback to the authors. We hope that the readers will find this book useful in their research. We gratefully acknowledge the support and encouragement received from Prof. Janusz Kacprzyk, the editor in chief of Springer series "Studies in Computational Intelligence", and Ms Heather King for the great assistance during the editorial work.

The editors also acknowledge that their research has been partially supported by the Spanish MICINN project "INCoS2009" (Ref: TIN2009-07489-E) and the IN3-UOC HAROSA Knowledge Community Program (http://dpcs.uoc.edu). Fatos Xhafa's work is partially done at Birkbeck, University of London, on Leave from Technical University of Catalonia (Barcelona, Spain). His research is supported by a grant from the General Secretariat of Universities of the Ministry of Education, Spain.

July 2010
Barcelona, Spain

Santi Caballé, Open University of Catalonia
Fatos Xhafa, Technical University of Catalonia
Ajith Abraham, Machine Intelligence Research Labs

Contents

III. Intelligent Secure and Resilient Networking Systems

List of Contributors

Ajith Abraham
Machine Intelligence Research Labs
(MIR Labs)
Scientific Network for Innovation
and Research Excellence
P.O. Box 2259
Auburn, Washington, USA
ajith.abraham@ieee.org

Pablo Arozarena
Telefónica Investigación y
Desarrollo, Emilio Vargas 6
28043 Madrid, Spain
pabloa@tid.es

Santi Caballé
Dept. of Computer Science
Open University of Catalonia
Rambla Poblenou, 156
08018 Barcelona
Barcelona, Spain
scaballe@uoc.edu

Alvaro Carrera-Barroso
Telefónica Investigación y
Desarrollo, Emilio Vargas 6
28043 Madrid, Spain
alvaroc@tid.es

Shinichiro Chaki
Network Service Systems
Laboratories, NTT Corporation,
3-9-11, Midori-cho,
Musashino-shi Tokyo, 180-8585
Japan
chaki.shinichiro@
lab.ntt.co.jp

Valentin Cristea
Strada Costache Marinescu No. 7,
sector 1
Bucharest, Romania
valentin.cristea@
cs.pub.ro

Simon Dobson
School of Computer Science
University of St Andrews
St Andrews KY16 9SX,
United Kingdom
sd@cs.st-andrews.ac.uk

Angela Fogarolli
University of Trento
Via Sommarive 14
38123 Trento, Italy
fogarolli@okkam.it

Akihiro Fujihara
Graduate School of Science and
Technology, Kwansei Gakuin
University,
2-1 Gakuen Sanda Hyogo 669-1337,
Japan
afujihara@kwansei.ac.jp

Javier García-Algarra
Telefónica Investigación y
Desarrollo, Emilio Vargas 6
28043 Madrid, Spain
algarra@tid.es

Sergio García-Gómez
Telefónica Investigación y
Desarrollo

Parque Tecnológico de Boecillo
47151 Boecillo (Valladolid),
Spain
sergg@tid.es

Benoit Gaudou
UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique (IFI)
42, Ta Quang Buu, Ha Noi,
Viet Nam
benoit.gaudou@alumni.enseeiht.fr

Bogdan Ghit
Strada Nerva Traian No. 12, Bl. M37, Sc. 4, Et. 4, Ap. 104, sector 3
Bucharest 031042, Romania
bogdan.ghit@cti.pub.ro

Kahina Hamadache
Orange Labs
42 Rue des Coutures
14066 Caen, France
kahina.hamadache@orange-ftgroup.com

Kenichi Higuchi
Network Service Systems Laboratories, NTT Corporation,
3-9-11, Midori-cho, Musashino-shi
Tokyo, 180-8585 Japan
higuchi.kenichi@lab.ntt.co.jp

Hesam Izakian
Department of Computer Engineering, University of Isfahan,
Hezar Jerib Avenue,
Isfahan, Iran
hesam.izakian@gmail.com

Angel A. Juan Perez
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Rambla Poblenou, 156, 08018
Barcelona
ajuanp@uoc.edu

Pavel Krömer
Department of Computer Science,
VŠB - Technical University of Ostrava, 17.
Listopadu 15, 708 33,
Ostrava-Poruba, Czech Republic
pavel.kromer @vsb.cz

Luigi Lancieri
Laboratoire d'informatique fondamentale de Lille (LIFL) Bat. M3 Université de Lille 1
59655 Villeneuve d'ascq Cedex,
France
luigi.lancieri@univ-lille1.fr

Daniel Lázaro Iglesias
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Rambla Poblenou, 156, 08018
Barcelona
dlazaroi@uoc.edu

Emerson Loureiro
Systems Research Group
School of Computer Science and Informatics
University College Dublín
Dublín 4, Ireland
emerson.loureiro@ucd.ie

Nicolas Marilleau
UMI 209 UMMISCO, Institut de Recherche pour le développment (IRD)
32, avenue Henri Varagnat,
93143 Bondy Cedex, Bondy,
France
nicolas.marilleau@ird.fr

Joan Manuel Marques i Puig
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Rambla Poblenou, 156, 08018 Barcelona
jmarquesp@uoc.edu

Hiroyoshi Miwa
Graduate School of Science and Technology, Kwansei Gakuin University,
2-1 Gakuen Sanda Hyogo 669-1337, Japan
miwa@kwansei.ac.jp

Max Mühlhäuser
TU Darmstadt - FB 20
FG Telekooperation
Hochschulstraße 10
D-64289 Darmstadt, Germany
max@informatik.tu-darmstadt.de

Paddy Nixon
Systems Research Group
School of Computer Science and Informatics
University College Dublín
Dublín 4, Ireland
paddy.nixon@ucd.ie

Kentaro Ogawa
Network Service Systems Laboratories, NTT Corporation,
3-9-11, Midori-cho, Musashino-shi
Tokyo, 180-8585 Japan
ogawa.kentaro@lab.ntt.co.jp

Jan Platoš
Department of Computer Science, VŠB - Technical University of Ostrava, 17
Listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
jan.platos @vsb.cz

Florin Pop
Soseaua Pantelimon No. 86, Bl. 409A, Sc. B, Et. 1, Ap. 51, sector 2
Bucharest 021637, Romania
florin.pop@cs.pub.ro

Marta Rybczynska
Faculty of Electronics and Information Technology, Warsaw University of Technology
Nowowiejska 15/19
00-665 Warszawa, Poland
marta@rybczynska.net

Dragos Sbarlea
Str. Capalna No. 10, Bl. 15A, Sc. 3, Et.2, Ap. 43, sector 1
Bucharest, Romania
dragos@rice.edu

Alina Simion
Aleea Stanila No. 3, Bl. H9, Sc. 2, Et.1, Ap. 27, sector 3
Bucharest, Romania
alina.gabriela.simion@rice.edu

Václav Snášel
Department of Computer Science, VŠB - Technical University of Ostrava, 17
Listopadu 15, 708 33,
Ostrava-Poruba, Czech Republic
vaclav.snasel@vsb.cz

Raquel Toribio-Sardón
Telefónica Investigación y Desarrollo, Emilio Vargas 6
28043 Madrid, Spain
raquelt@tid.es

Xavier Vilajosana Guillen
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Rambla Poblenou, 156, 08018 Barcelona
xvilajosana@uoc.edu

Tuong Vinh Ho
UMI 209 UMMISCO,
Institut de la Francophonie
pour l'Informatique (IFI)
42, Ta Quang Buu, Ha Noi,
Viet Nam
ho.tuong.vinh@auf.org

Stefan G. Weber
TU Darmstadt - FB 20
FG Telekooperation
Hochschulstraße 10
D-64289 Darmstadt, Germany
stefan.weber@cased.de

Fatos Xhafa
Dept. of Languages and
Information Systems
Technical University of
Catalonia
Campus Nord. Ed. Omega
C/ Jordi Girona, 1-3
08034 Barcelona, Spain
fatos@lsi.upc.edu

Wikipedia as a Source of Ontological Knowledge: State of the Art and Application

Angela Fogarolli

Abstract. This chapter motivates that Wikipedia can be used as a source of knowledge for creating semantic enabled applications, and consists of two parts. First, we provide an overview over different research fields which attempt to extract knowledge encoded by humans inside Wikipedia. The extracted knowledge can then be used for creating a new generation of intelligent applications based on the collaborative character of Wikipedia, rather than on domain ontologies which require the intervention of knowledge engineers and domain experts. Second, as a proof of concept, we describe an application whose intelligent behavior is achieved by using Wikipedia knowledge for automatic annotation and representation of multimedia presentations.

Keywords: annotation generation, unique identifiers, automatic WSD.

1 Introduction

The state of the art in information retrieval shows that content which has not been specifically produced with later retrieval in mind creates a notable negative impact on the qualitative performance of retrieval systems. This is a fact that we all perceive in our daily work with Internet search engines, the performance of which is especially problematic on the content we will be dealing with in the second part of this work, namely different kinds of unstructured or semi-structured information sources.

One way to improve this situation is to make content better suitable for later retrieval, by equipping it with a semantic description. This can be achieved by the use of a combination of several state-of-the-art techniques from the fields of Information

Angela Fogarolli
University of Trento, Via Sommarive 14, 38123 Trento, Italy
e-mail: fogarolli@okkam.it

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 1–26.
springerlink.com

Extraction an Natural Language Processing, and by relating their results to semantics provided by background knowledge. An issue in this context is the mismatch between diversity of content (spanning more than one domain) that is often encountered, and a common lack of ready-to-use conceptualizations which are suitable to describe such content. Additionally, domain ontologies require the intervention of knowledge engineers and domain experts, which is not desirable or even affordable in some cases.

One possible approach to reach genericity and domain independence is by using Wikipedia[1] as a source of knowledge instead of domain ontologies. The reason why Wikipedia is particularly suitable for such applications arises from several angles: first, articles are collaboratively written by humans, and typically through many revision stages. As a result of this process, an article is a description of a topic and represents an agreement among authors. The synergies inside the Wikipedia user community can be represented by a Participatory Network [26], which represents a system of exchange and integration. Participatory Networks are based upon conversation Theory, and the basic concept is that knowledge is created through conversation. In the case of Wikipedia, conversation is activated through new web-based collaboration technologies. Secondly, the production of Wikipedia articles follows Pask's conversational theory [38]. Pask defines the network and its stability through cybernetic theory, which explains that when an agreement based on understanding is reached between two people even if the agreement was built upon transient element which changed, they can come to another agreement based on understanding which will replace the old one. These facts support the frequently taken approach to use Wikipedia as a source of background knowledge.

The rest of this chapter is structured as follows: Section 2 starts with an introduction to semantics and ontologies, and motivates the use of Wikipedia as an ontological resource. Section 3 provides an overview of the state of the art in extracting semantics from Wikipedia content and its structure. Finally, we illustrate how Wikipedia semantics extracted in various ways can be used as ontologies in creating intelligent applications. A proof of concept of such an application is described in section 4, where an approach based on Wikipedia semantics tries to contribute to the state of the art in content-based multimedia indexing.

2 Semantics and Ontologies

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. In computer science, ontology is, in general, a representation of a shared conceptualization of a specific domain, for a more detailed description refer to [10]. Ontology represents concepts, a concept is a agreed meaning of a term.

[1] Wikipedia is a freely available online encyclopedia which is constantly growing in size and in fame thanks to the copyleft license that allows the content to be copied, modified and redistributed as long as there is an acknowledgment of the author and the new content is published under the same license. See http://www.wikipedia.org

Philosophers define the meaning as something which remains unchanged during the process of translation in a different language while the receivers of the information containing the term are able to refer to the same original thing.

In natural language a concept is associated with an idea characterized by a certain mental state during the usage of a term. Ontologies are a representation of entities which are real world concepts. Klein and Smith suggests in [18] that an entity can be of two kinds: an instance called individual in the ontology literature or a type. Other ways to call a type can be 'class', 'category', 'kind', 'genus','species' or 'taxon'. A golden retriever is an instance of type dog. They also classify entities first on their temporal nature: occurant in the case of process and events or continuant which state can be persistent in relation with time. Continuant entities can also be variable independent in this case are sub-classified in material or immaterial or their status can be dependent such as temperature, quality or functions.

Ontological classification works well in restricted domains [11], where there are formal categories, stable and limited entities. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate the act of drawing conclusions(inferencing).

In Knowledge Representation the debate is whether the vocabulary and the structure for representing knowledge must come from experts as in the Ontology or taxonomy area, or whether it should come from a community effort of open and self organized content as folksonomies. Ontology represents entities and their relationships while a taxonomy is the grouping of organisms into genus and species. Both of them are used for representing knowledge. Folksonomies are based on user tagging of resources. Tags allow multiple prospectives founded on different social characteristics of the people involved in the process, this could lead to a more global and shared view as opposed to Ontologies which aim to do that but are usually developed by a restricted number of experts.

Gruber 2005 [19] bridges Ontologies and folksonomies proposing a conceptualization of the activity of tagging called TagOntology. TagOntology stores knowledge about an object been tagged, a tag, a tagger and the information source. This could allow reasoning about tags such as this resource could be also tagged using this synonym. TagOntology's aim is to enable the exchange of tag data and the building of tagging systems that can compositionally interact with other systems.

Semantic Web technologies aim to annotate documents based on domain Ontologies. In this way the semantic of the produced annotations are well defined. Semantic annotations define in a formal way concepts and relationships between them. The use of knowledge embodied in annotations is being introduced in diverse domains.

A big problem in dealing with Ontologies is the domain coverage. It is in fact impossible to find a broad, domain-independent collection of individual terms (as opposed to concepts) which are connected by relations. One cause of the slow adoption of Semantic Web technologies is due to not proper domain coverage and to the difficulties in defining or using Ontologies.

We explored the possibility of using SemanticWeb ontologies for annotating multimedia material and for discovering and presenting to the user relations between the searched topics and other topics, based on the relationships between entities in one or more domain ontologies. We also experienced difficulties in finding ontologies which cover a variety of domains, since a multimedia presentation (i.e a e-Lecture) can cover an unpredictable amount of domains (e.g. computer science, history, meteorology, geography, math...). In addition, the terms expressed in the presentation are usually individuals of an ontology (e.g. the term Collection in a Java Programming class could be modeled as an instance of a data container class in a Java Programming ontology) and finding populated ontologies with a wide coverage of individuals to date is a big challenge, and usually requires the involvement or a knowledge engineer.

Using Wikipedia as a source of knowledge instead of Semantic Web Ontologies have the advantage to be domain-independent, very extensive and could provide a way for enabling intelligent behavior.

Wikipedia contains a classification of topics, organized with an hierarchy of categories and with relationships between elements. One advantage of using it is that the social collaborative network around it makes its content always up to date and it covers in detail a huge amount of topics in different domains and languages. In addition it also takes into account the different possible meanings of a term through a disambiguation page.

2.1 Wikipedia as an Ontological Resource

The view of Wikipedia as an ontological resource is recent but there is a lot of research in this area. Wikipedia category structure is equivalent to a simple ontology [64] with a finite controlled vocabulary, an unambiguous interpretation of classes and relationships and an hierarchical relationships between classes.

According to Obrst's definitions in [36], Wikipedia not only offers *weak semantic* information, such as parent-child relationships, but it also contains lexicographic relationships that – once the domain of interest is determined – can offer *medium semantic*. In Wikipedia we do not have *strong semantic*, i.e. we can not describe real-world relationships such as "a car has a minimum of four wheels" as with the usage of an ontology. We can only deduce that concepts are connected without knowing how; we can tell that one concept in one category is related to other concepts which are linked in the description of the concept itself.

In Wikipedia, the concept of class and instance are not separated as in the ontological sense, due to the fact that it is not constrained to a formal model, for the reason of which it is not possible to formalize reasoning on the Wikipedia content directly.

Based on that Yu et al. [64] apply some state of the art evaluation criteria for Ontologies to the Wikipedia category structured in computing a browsing task. The experiment run with 18 volunteers confirmed that a simple ontology as the Wikipedia category structure can help browsing. The authors show also that an untangled version of Wikipedia where multiple parents of a given category had been removed, performed even better. Pozzetto in [39] argues that Wikipedia, as it is, can not be considered an ontology because the relations between categories do not have an explicit semantics such is-a or part of, and justifies his work on extracting relationship between categories [40].

In [16] Wikipedia is used for ontology construction where domain specific corpus are usually required for building a corresponding concept hierarchy. For that purpose category information are used to create a graph connecting a set of pages in a domain. For determining which pages are domain relevant a threshold is calculated considering in and out links. Since a page can belong to multiple categories the more out-links of a page are pointing toward other pages in the classification tree the most the page is likely to be domain specific.

Building domain specific thesauri is costly and high maintenance is necessary. Milne et al. [33] have investigated the usage of Wikipedia as a thesaurus. The investigation is about how Wikipedia structure is compatible with a thesaurus structure for the reason that each article describe a single concept, the title is descriptive and short and the hyperlinks between articles cover many of the types defined in the standard for thesauri(ISO 2788) in particular equivalence, hierarchical and associative relations. Moreover a thesaurus should be a controlled vocabulary to bridge different senses and synonyms. In Wikipedia disambiguations pages act on the same way. Wikipedia also provide hierarchical relations reflected by the category structure and associative relations are expressed trough hyperlinks between pages.

In [33], Milne et al. have also evaluated Wikipedia coverage and accuracy of the terminology and relations against a domain specific thesaurus in agriculture. The surprising result is that – even tough Wikipedia was not envisioned to be used as a thesaurus – it outperforms a tailored thesaurus in supporting the analysis of a domain-specific document collection.

Siorpaes and Hepp in [45] and Hepp et al. in [20] propose the usage of Wikipedia URIs as an identification of resources for knowledge management. They combine the Wikipedia URIs for an entity with properties and relations expressed in other metadata standards such as Dublin Core, with textual description or using relationships defined in external ontologies. Furthermore the analysis positively evaluated Wikipedia content with regard to its quality as a vocabulary for annotating Web resources.

The use of Wikipedia URLs for concept identification can solve the identity mismatch and guarantee interoperability between domain ontologies. While the extensive ongoing research effort for extracting an ontological view from Wikipedia could also provide a reference ontology for many domains.

3 Extracting Semantics from Wikipedia: Related Work

In recent years Wikipedia has become a valuable resource for both the Natural Language Processing (NLP) community and the Information Retrieval (IR) community. Wikipedia quality seems to be comparable to other encyclopedia, A Nature article [17] from 2005 found that the number of errors in Wikipedia are not significantly different form those in Britannica. In [52] a study about the quality of Wikipedia articles is discussed. Based on quality parameter measurements such as authority/reputation, completeness, complexity, informativeness, consistency, currency and volatility, it could discriminate high quality articles in the Wikipedia collection. Semantic coverage of Wikipedia articles is a good aspect for studying the quality of a community process involved in Wikipedia.

Since popular articles are going through a process of discussion and repeated editing before reaching a stable state, the authors [51] investigated this aspect and suggested a way to assign reliability measure to articles based on the semantic distance between revisions. The hypothesis is that a document is mature if despite ongoing changes it is semantically stable and revision on stable documents do not make significant changes. On the other hand another study [21] shows that user interaction data itself are not sufficient for judging article quality while article length appears to have some merits in identifying quality articles.

Dictionary and encyclopedia knowledge might complete each other and provide knowledge missing in corpora [41]. In [28], Wordnet is used for word disambiguations and for query expansion. After the sense of a query word is determined, its synonyms, words or phrases from its definition, its hyponyms and its compound words are considered for possible addition to the query.

Since Wikipedia contains a vast amount of information there have been two main approaches to make Wikipedia machine readable; one is to embed semantic notation in its content. The second is to automatically extract information from Wikipedia content based on its structure and content.

The first kind of approach is about coding semantic notations in the Wikipedia article itself for creating a collaboratively taxonomy which is called Folksonomy [35]. Folk-Ontologies are a way to combine the collaborative nature of tagging content of Folksonomy with a more rigorous ontological classification. Folk-Ontologies focus on specific types of relationships between things, inviting the Wikipedia authors to add this kind of semantic tags to their article, approaches of this kind can be found in the SemanticWikipedia project [55] and in IkeWiki [43].

MediaWiki [9], the wiki software behind Wikipedia, itself enables authors to represent structured information in an attribute-value notation, which is rendered inside a wiki page by means of an associated template.

The second main stream of Wikipedia related work is on automatic knowledge extraction from Wikipedia content. The explanation of this part of work is organized in the followng sub-sections. Each sub-section is focused on the investigation of a particular aspect or outcome of the automatic knowledge extraction problem:

- *Wikipedia Knowledge Bases* describes approaches for mapping the Wikipedia content and structure into knowledge bases where the content is organized and retrieved using Semantic Web technologies, those knowledge bases usually contains only a subset of the information provided by Wikipedia.
- *Category and Link Structure Analysis*, this aspect is about extracting semantic information from the category and link structure of Wikipedia.
- *Relationship Mining*, is another important aspect for creating an ontological view of Wikipedia. This section describes the work in finding relationships between entities and concepts mentioned in the text of Wikipedia articles.

- *Text Categorization* section focuses on the strategies used for understanding high level categorization for the semantics included in Wikipedia articles. This techniques are essential for mapping the content into high level ontological classes and relationship among them.
- *Natural Language Processing* , *Article Summarization*, *Word Sense Disambiguation* and *Semantic Relatedness* sections are also tools for extracting different type of semantics that can enable the creation of an ontological view of Wikipedia and semantic aware applications.

3.1 Wikipedia Knowledge Bases

DBpedia [3] is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows to ask sophisticated queries against Wikipedia and to link other datasets on the Web to Wikipedia data.

In March 2010, the DBpedia dataset describes 2.900.000 ”things”, including at least 282.000 persons, 339.000 places, 88.000 music albums, 44.000 films. It contains 807.000 links to images, 3.840.000 links to relevant external web pages and 4.878.100 external links into other RDF datasets (Geonames,Musicbrainz, WordNet, World Factbook, EuroStat, Book Mashup, DBLP Bibliography and Project Gutenberg datasets). Altogether, the DBpedia dataset consists of around 479 million RDF triples. DBpedia extracts [4] RDF triples from Wikipedia informations presented in the page templates such as infoboxes and hyperlinks.

Yago [48] is a knowledge base which extends the relationships of DBpedia extending the standard RDF notation. The last downloadable version at the time of this writing is dated December 2008 contains over 2 million entities (like persons, organizations, cities, etc.) A YAGO-query consists of multiple lines (conditions). Each line contains one entity, a relation and another entity.

Due to the large data schema of these two initiatives, users can hardly know which properties and identifiers are used in the knowledge base and hence can be used for querying, so the user must be guided in the process of query creation suggesting step by step alternatives. For the reason of which creating automatic methods for querying such as knowledge base is not an easy task where is unknown how concepts and relationships are labeled in the ontology.

3.2 Category and Link Structure Analysis

Wikipedia categories are widely analyzed as a semantic source. Wikipedia’s category structure is a thesaurus [57] which has been collaborative developed and used for indexing Wikipedia articles. Categories are hierarchically organized into sub and super categories. The category structure is not a tree, some categories have multiple super-categories and an article can belong to multiple categories.

Synarcher [25] is another work based on Wikipedia knowledge which searches for synonyms and related terms in the Wikipedia category structure and analyzing hyperlinks between pages. The algorithm could be used to extend queries in a search engine, or as an assistant for forming a dictionary of synonyms. Another work which explores categories in Wikipedia is the one of Chernov et al. [5]. The authors suggest that semantic information can be extracted form Wikipedia by analyzing the category structure and they propose a way to calculate a connectivity ratio which correlates with the strength of the semantic connection among them. Wikipedia categories are also used for document classification by Schonhofen [44]. Schonhofen tries to find the Wikipedia categories that most characterize a document, by matching Wikipedia article titles with words of the document to be classified. Categories are then ranked based on some factors, including the number of shared words between the document and a Wikipedia article title and number of Wikipedia articles with similar titles. Thom et al. [54] explored Wikipedia category structure to improve entity ranking effectiveness. Watanabe et al. present another work on Name Entity categorization [61] based on category information extracted from the linked HTML text in the articles.

Syed et al. in [50] describe an approach for identifying topics and concepts associated with a set of documents. The approach is based on the Wikipedia category graph for predicting generalized concepts and uses article links to help predict concept when an article is not associated with a specific category.

The link structure in Wikipedia draws a huge network between pages which facilitate the navigation and the understanding of concepts.

Adafre and de Rijke [2] in 2005 as first analyzed the link structure in Wikipedia. They tackle the problem of missing links between articles. For doing this they cluster similar pages based on similar link structure and then they examined these cluster to find missing links between them.

Link structure in Wikipedia has also been studied by Voss and it can be described as a power law function. In [56] the author explains how the exponential growth of links and the assumptions that the more a non-existing article is linked, the more likely someone will create it.

Fogarolli and Ronchetti [13] relies on strong links which are bidirectional links between pages and interlanguage links for multilingual domain disambiguation and for content summarization by mapping concepts extracted from a text to Wikipedia definitions.

Kamps and Koolen [24] examined Wikipedia link structure and stated that link structure is an indicator of relevance especially if considering links between pages retrieved in response to a search request. In other words links can help defining a context and can improve performance in information retrieval.

Hyperlinks structure in Wikipedia is also used for calculating related pages to an article. Ollivier and Senellart [37] process these relationships using Green Measures which is a function introduced in electrostatic theory for computing the potential created by a charge distribution. Green measures are applied as a finite Markov chain to a graph modeled by hyperlinks among Wikipedia articles.

3.3 Relationship Mining

Culotta et al. [8] deal with the problem of extracting relations between entities from Wikipedia English articles. The method mines key patterns from the syntactic and semantic structure to measure similarity between entity pairs rather than using only lexical information. The method requires a manual training. The domain is restricted to thirteen relationships between entities. This methodology is similar to the one used by Wu and Weld in [63] for the fact that both approaches analyze structured and semi-structured information. The first approach is more concentrated in relationships between concept while the latter extracts attributes related to the concepts. Nguyen et al. [34] describe a similar work on relationship extraction from Wikipedia based on sub-tree mining. Entities from each articles are semi-automatically selected, then a dependency tree is calculated with the aim of extracting the most important relationships among a pair of entities. Another approach which requires training is described by Wang et al. in [58]. This method is called PORE (Positive-Only relation Extraction) and aims to extract and classify relations instances from Wikipedia text by identifying strong negative examples from unlabeled data and then iteratively classifies more negative data until convergence. The method iterates until no positive data can be generated anymore.

3.4 Text Categorization

E. Gabrilovich and S. Markovitch [14] intent to categorized text by mapping the most frequent concepts in text to Wikipedia articles. They use a classifier for mapping articles to Wikipedia definitions. Their classifier is also based of TF-IDF[2]. From the reported experiments is clear that their algorithm needs a big amount of training data to startup the process. Based on the Wilcoxon test[3], their Wikipedia-based classifier is significantly superior to the baseline. These findings confirm that encyclopedic knowledge should be particularly useful when categorizing short documents, which are inadequately represented by the standard bag of words.

Annotations are also used for categorizing content, Siorpaes and Hepp [45] proposed a manual annotation of Wikipedia and other resources based on Proton Ontology[4]. In the game, they show the first paragraph from a Wikipedia page. Then, they ask the user to select whether this Wikipedia entry describes a set of objects (i.e., a class) or a significant single object (i.e., an individual). The evaluation focuses on the analysis on the number of correct choices and on the types of mistakes that were observed.

Regarding Wikipedia annotations, other interesting works are the one of Ramos and Warner [29] and Atserias et al. [23] where Wikipedia is automatically annotated using three taggers (SuperSense Tagger [6]) trained on three different corpora. Having a tagged reference corpus would be very useful to homogenize experiments

[2] http://en.wikipedia.org/wiki/TF-IDF

[3] http://www.stats.gla.ac.uk/steps/glossary/nonparametric.html

[4] http://proton.semanticweb.org/

to make results comparable however no measure of performance are told in the papers.

Sangweon et al. [49] show an attempt of extracting what they called "common sense knowledge" from Wikipedia articles. Basically they extract RDF triples encoding the subject and object of a sentence as a generic "rdfs:Class" notation, the predicate as an instance of "rdf:Property" as they are written in the text. In their evaluation was assessed wheather the extraction made on two-hundred pages preserved the meaning of the original page by the judgment of two evaluators.

Zesch et al. [66] underling the potential of using Wikipedia for natural language processing and compare it with conventional resources, such as dictionaries, thesauri, semantic Wordnet. They provide an API for accessing Wikipedia content and structure.

3.5 Natural Language Processing

Natural Language processing technique are wildly used in discovering semantics in Wikipedia text. Zesch and Gurevych in [65] compute a graph-theoretic analysis of Wikipedia Category graph and show that it can be used in NLP like other semantic networks such as WordNet. In [42] a methodology for semi-automatic discovering of lexical patterns that allows discovering of relationships between concepts is presented. In this paper, Ruiz-casado at el. try to generalize patterns in sentences which contain the same concepts by using wild cards or a vertical bar to separate adjectives that can be used as alternatives between two concepts. The approach has been evaluated on a very small subset of Wikipedia. The considered articles cover seven subjects. The precision has been estimated manually checking only the last fifty relationships extracted. There is a big difference on the precision value depending on the test subject. Precision goes from 7% in the case of player-team to 90% for the death-year.

ISOLDE [62] is a system for deriving a domain ontologies using named-entity tagger on a corpus and combining the extracted information with Wikipedia and Wiktionary. The results show that this kind of approach works better with semi-structured information such as dictionaries.

KYLIN [63] tries to automatically complete the information presented in the Wikipedia infoboxes analyzing and disambiguating text and links in Wikipedia pages.

3.6 Summarization of Wikipedia Articles

Adafre et al. report in [1] about different techniques for summarizing Wikipedia content, using different algorithms for each sub-task. In summarizing the article the authors take also into account snippets of text from other Wikipedia articles that can be important for rendering a complete picture of the topic. For each article the relevant sentences are identified and ranked using a score value. The process concludes removing sentences which repeat the same concept. For identifying relevant sentences the approach also extracts sentences from articles anchored through hyperlinks. This is done only if the anchored definition contains the entire or part of title of the to-be-summarized article. The importance of the sentences is estimated confiding in the position of the sentence in the article. The reason of this choice is based on the fact that Wikipedia guidelines suggests that sentences which come first in the article are more important. The similarity among sentences describing the same concept is calculated combining category information and the word overlap between documents.

3.7 Word Sense Disambiguation

Wikipedia as been also investigated as a source of sense annotations for Word Sense Disambiguation(WSD). WSD task, given a set of sense-annotated example can learn a disambiguation model that can predict the correct sense for future occurrences of a word. In other words, WSD aims to assign dictionary meanings to instances of a corpus. Approaches to WSD are of three kinds: the first is the Knowledge-Based Disambiguation approach, which uses some external linguistic resources such as dictionaries and is based on heuristics and analysis of the context where the ambiguity is found. A second approach is called Supervised Disambiguation which employs training data, manual annotations, feature extraction to be used by classifiers. The latter is the Unsupervised Disambiguation approach which focus on not labeled corpora where sense example sets do not exist where for feature analysis and clustering are applied.

In WSD the Knowledge-Based approaches are many. There are methods like LESK algorithm[5] which calculates overlap with respect to dictionary definitions. A second method is based on similarity measures computed on semantic networks while another approach is focus on selectional preferences which will help deriving the meaning based on the sense of other words in a given context.

An implementation similar to the Knowledge-Based approach is mentioned in [12] which proposes an unsupervised method based on examples of usage of the ambiguous term embedded in the article itself for word disambiguation.

Mihalcea in [30] and [31] discuss the use of Wikipedia for WSD. In [30], the author reports about the use of Wikipedia content for avoiding the bottleneck in WSD of not having enough examples of a term usage. In her approach, she selects all paragraphs in Wikipedia which contain a contextualized reference to an ambiguous term in the link label and then maps the different Wikipedia annotations to word senses instead of relying on the Wikipedia disambiguation pages. This is due to the face that sometimes not all meaning are elicited in the disambiguation page. Finally, the labels which describe the possible senses for a word are manually mapped to WordNet senses. In this way the number of example for each word can increase improving the performance of a classifier. In her second work [31], Mihalcea describes an use case of her WSD algorithm to an application which associate terms

[5] The Lesk algorithm is a classical algorithm for word sense disambiguation introduced by Michael E. Lesk in 1986. For more details refer to [27].

in an input text to Wikipedia definitions. The keyword extraction from the text is done using a controlled vocabulary. WSD is done in three different ways. Using a Knowledge-Based calculating the overlap of the Wikipedia definition with the paragraph where the text occurs (similar to Lesk algorithm). A second approach that has also been tested in [31] is a data-driven method which use a machine learning classifier, giving as a training all the occurrences where the word is found in the link plus all the possible Wikipedia definition articles which represents the possible meanings. Additionally they experimented also a combination of the first two approaches.

Cucerzan in [7] domain disambiguation approach is focused on the extraction from Wikipedia pages of different kind of knowledge. The first kind is called surface form, which includes the extraction of entities from titles of pages and links. The second type of knowledge derives form the calculation of tags based on category information and the domain of the definition (what is usually written in brackets in the page name). The disambiguation process maps corpus entities to all possible disambiguation taking into account surface form and category tags for creating a document vector that will be compared with a Wikipedia entity vector for all possible entity disambiguation pages. The disambiguation is done by maximizing the similarity between the document vector and the vector of the Wikipedia article which combines category and lexical information (entity vector). The evaluation has been done manually against a set of news stories with a reported high accuracy of 91%.

3.8 Semantic Relatedness

Ponzetto and Strube in their works have explored Wikipedia structure for extracting semantic information. In the WikiRelate! paper [47] they use Wikipedia to compute semantic relatedness between two terms. Giving any two words semantic relatedness is calculated between a pair of articles which contain those words in their titles. The computation takes into account word-based similarity and the distance between the articles' categories based on the Wikipedia category tree. In a subsequent work Ponzetto and Strube [40] aim to extract from Wikipedia a taxonomy containing a large amount of subsumptions, i.e. is-a relations this work could lead to construction of a knowledge based of Wikipedia concept [39].

E. Gabrilovich and S. Markovitch proposed in [15] the ESA (Explicit semantic analysis) technique to represent content by means of Wikipedia concepts. The concept are selected and ordered using text classification techniques, vector space modeling, similarity measures and machine learning algorithms to represent the meaning of a text fragment as a weighted vector of concepts derived from Wikipeda. Relatedness are calculated through the comparison of concept vectors.

An approach similar to ESA is presented by Milne [32] in his technique only Wikipedia links are used in calculating similarity rather than the entire content however accuracy it is much worst than ESA and WikiRelate!

Janik and Kochtut [22] also work on text categorization using Wikipedia as an ontological resource. Their proposed approach tries to find a semantic similarity between a document and a certain category described by an RDF ontology derived from Wikipedia. For achieving this they transform each document in a semantic graph which is used to calculate similarity to the categories defined in the ontology.

4 An Intelligent Application of Wikipedia Knowledge

This section describes a proof of concept implementation of the extracted Wikipedia Knowledge in the area of content-based indexing of multimedia and information retrieval. The purpose of the implementation is to show how based on Wikipedia knowledge is possible to automatically offer an intelligent representation of multimedia presentation content which will enable their intelligent treatment.

Multimedia information retrieval is a heterogenic field [59]. Unlike many other research fields, a wide range of data types have been studied by researchers. Multimedia resources include text, hypertext, audio, graphics, animation, image, video, rich text, spread sheet, presentation slide, combinations of these. The resource panorama is also diverse, the research in the field spread around different approaches in different prospective such as system, content, services, user interfaces, social/business and application. If from one prospective diversity helps to enrich the possibility of innovation, on the other hand there is a danger that some work would not be considered because it does not conform with the common state of the area approaches recognized by the main players and this could be a potential threat for innovation [60].

The challenge in multimedia content research is to understand the multimedia content in order to properly organize it for facilitating its retrieval.

A common practice is to associate to documents a list of metadata which briefly described their content and will enable intelligent manipulation of the multimedia resource. The description should have a high level of semantic in order to be used for answering human needs of classification and retrieval. The research in this field is mainly focused on feature extraction from key shots which enable a better understanding of the scene. During this process, low level features such as text, colors, shapes and segments are evaluated, which can contribute to the understanding of the meaning of a picture. Metadata can be manually generated but of course it is costly, time consuming and prone to errors. Also the agreement between annotator can notable differ and requires domain expertise. Since the amount of documents people are dealing with are constantly increasing, manual annotation is not a viable solution while knowing what a document is about it is of fundamental importance for effective knowledge management.

Automatic or semi-automatic technique can be implied instead as an alternative to human annotations. The limitation of automatic annotations is low recall when some annotations are missing or low precision when the annotation are inaccurate. A problem about automatic annotations is the extraction of relationships [53] between them. However, annotations alone do not establish the semantics of the vocabulary used in the annotations.

Multimedia presentations are an example of resource where more than one presentation modality is used in combination. A multimedia presentation is usually composed of a video with focus on the speaker, presentation slides and other textual documents which can be identified by the presenter as related source of information.

In accordance with Snoek and Worring [46], in the approach described in more details later in the section, modalities are combined since the ability of expressing semantic in a video document is explicated using at least two modalities for extracting semantics out of it. For this reason, the indexing and description of multimedia presentation focus not only on the video itself but also on the related material that comes with the video presentation, such as presentation slides, whiteboard recording and reference documents, semantically coupled and time-synchronized with the video content. Since this collection of information sources is relevant for improving the knowledge transfer between the speaker and the audience, and thus has to be taken into account in combination.

The approach used for this use case focuses on auditory and textual modality instead of visual modality. Unlike other domains such as movie or news, in video presentations the images in the keyframes are more or less still, usually the speaker and part of his/her presentation is captured. The scene almost never changes, the transitions being related to a change of focus (from slide to blackboard and back) or to the change of slide.

The presented approach focused only on auditory and textual modalities, even though low level feature recognition such as gesture and facial prosody might give information about importance of certain passages, this issue would only bring a minor added value in comparison to the knowledge that can be retrieved exploring the auditory and textual modality. Furthermore, relating the extracted speech and textual content with the right domain knowledge could provide another mode to tackle the semantic gap allowing more effective classification and searches on the video content.

Hence, this study investigates textual modality analyzing the full content of the related material such as slides or documents and the auditory modality translating it in textual since textual modality represents the most promising aspect of the data we can process. Automatic speech recognition (ASR) is applied to the video soundtracks and subsequently STT translation (speech to text) provides data that can be analyzed in combination with the other textual resources such as slides, notes and other documents.

In a multimedia presentation the video with the speaker appears together with presentation slides or additional notes. Video and slides are synchronized and can be navigated by means of a temporal bar or by slide titles.

4.1 A Semantic Discovery and Representation Approach

This section describes the methodology for extracting semantic annotations from multimedia presentation content based on Wikipedia knowledge. Those annotations express the most important concepts in the presentation content and their relationships.

As showed in figure 1 the process for intelligent describing the multimedia presentations is divided in two blocks. The first one deals with content analysis, the speech is converted into text using an out of the box automatic speech recognition(ASR) tool, then the transcriptions is combined with text extracted from slides or other documents which are part of the presentation. All the presentation content is then treated in its whole. During the content analysis phase the major concepts are identified in the material and mapped to Wikipedia taxonomy. Then based on the strong link analysis combined with lexicographic relationships between Wikipedia articles relationships between concepts are extracted.

The second module is responsible for representing the semantic annotations though Semantic Web technologies, in our example RDF format[6]. To have such a representation increases possibility of interoperability and enables semantic search.

A more detailed explanation of the process is given in the next paragraphs.

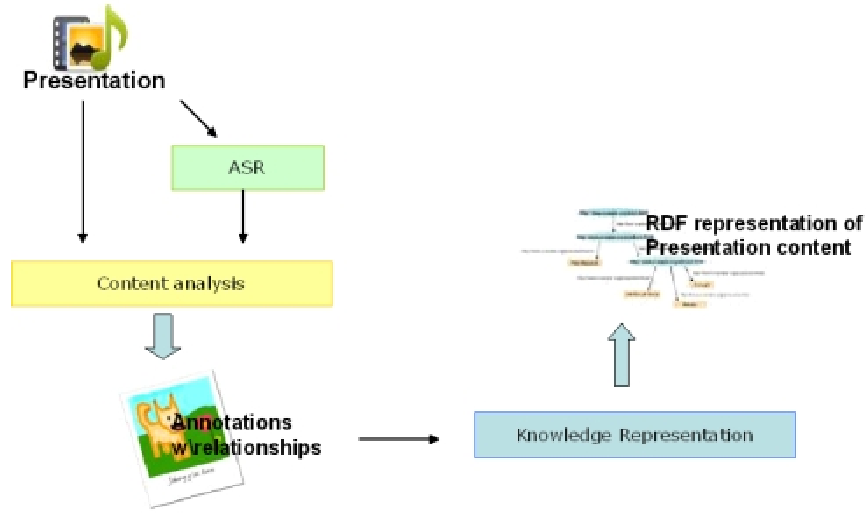


Fig. 1 Architectural overview

Content Analysis and Annotation Generation

Information Extraction output is a term vector based on TF-IDF weights for every presentation which consist in the fifty most important terms for the presentation material which comes from the combination of the transcript text, presentation slides and other textual resources the speaker points to.

The words in the term vector have been stemmed and are extracted combining all the presentation material. The process takes into account every term in the presentation term vector. The goal of the process is to find a Wikipedia page for every word

[6] http://www.w3.org/RDF/

in the term vector which has a definition that semantically matches the meaning of word in the presentation. For archiving this goal, for every term in the term vector, first is performed a look up all the Wikipedia page titles which starts with that term. At the end of the process is possible to associate to each of the n-most important terms of the presentation a summarized Wikipedia definition. In this way the annotations obtained through Wikipedia describe the content of the presentation. In the multimedia presentation use case for example only the first six most important annotation are displayed.

Next, page disambiguation is calculated for every term in the term vector, since the different senses of word are represented in Wikipedia through a disambiguation page. Each article in Wikipedia is identified by its title. The title consists of a sequence of words separated by underscores. When the same concept exists in different domains that name is concatenated with a string compose by a parenthetical expression which denotes the domain where the word has a specific sense. For example a Wikipedia query for the word ”Collection” returns the Wikipedia disambiguation page Collection, which points to other pages such as Collection(horse), Collection(museum), Collection(Joe Sample album), Collection(agency), Collection (computing), Collection_class. The string between parentheses identifies the domain. In order to choose which is the right definition to be picked for the presentation domain, we proceed analyzing the hyperlinks present inside the pages of all the possible candidate definitions listed in the Wikipedia disambiguation page. For each candidate definition we consider only its strong links.

A page has a strong link with page P_d if in P_o exists a link to P_d and in P_d there is a link back to P_o.

$$P_o \longleftrightarrow P_d \tag{1}$$

Hence, strong links represents the most important connections between topics. For example, the topics “Athens” and “Greece” are strongly linked since in the Wikipedia page about Athens is written that it is the capital of Greece, while the page on Greece reports that Athens is the capital of the state. A minor town located in Greece, such as Florina will instead have a “weak” link with Greece, since in its page it will be declared that the town is in Greece but in the page for Greece the minor town will most likely not be mentioned. For more information about the strong link theory and the disambiguation process please refer to [12]. For every term in the presentation, strong links are taken into account for computing the disambiguation process.

The best definition among the candidates is the one having the majority of words in the presentation material in common with the article name anchored from a strong link.

Since words in a term vector are stemmed, the strong links must be stemmed as well before comparing them with the keywords in the term vector. The expected result of the process is a complete disambiguated term vector for the presentation material. Every disambiguated word is mapped to a Wikipedia definition page, which describes the meaning of the word in the corpus.

Discovering relationships between annotations

The relationships between the concepts identified through the text analysis process are extracted using Natural Language Processing techniques on the Wikipedia article. In the previous section we described that we can map each concept in the multimedia presentation content to a Wikipedia article. The relationship between two concepts in Wikipedia are drawn by means of strong links. We mind the Wikipedia article text where a strong link is found and to extract the predicate between a subject which is the main topic of the article and another topic defined by a strong link. The connecting predicate between subject and object will act as label of the relationship between the two. The majority of the relationships we extracted are is-a kind but we could also have others; in this case we use the predicate in the infinitive form for labeling the relationship.

Knowledge representation of multimedia presentation content

In the presented use case, multimedia presentations are described using semantic annotations and their relationships. In particular in our approach concepts in the material are mapped to Wikipedia articles. Every concept extracted from the multimedia content is represented as an instance of a class called "article" which describes, using a Wikipedia definition, the concept itself.

An example of a RDF representation between the concept Map, HashMap, List, ArrayList, Set and Tree extracted from a multimedia presentation about Java Programming follows.

The related RDF graph created with a in-house-built tool giving the RDF text presented in this section is shown in figure 2.

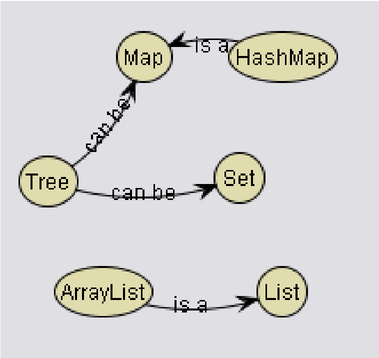


Fig. 2 Example of a RDF representation of multimedia content

Wikipedia URIs are used as resource identifiers. Any arbitrary number of attributes can be associated to a resource, in the example we define only the attribute name and the identifier for each resource but many others could be added in the same way. The predicates which hold a relationship between article resources are derived from the lexicographic relationships extracted during the text analysis of each Wikipedia article. All predicates are listed in the RDF schema file (RDFS). A RDF representation allows us to describe multimedia content and provides as well an extensible formal representation.

In the following paragraph is shown an example of a partial RDF representation of the content of a multimedia presentation about Java Programming.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-
    rdf-syntax-ns#'>
  <!ENTITY wiki 'http://mydomain.
  org/wiki-  rdfs#'>]>
<rdf:RDF
    xmlns:wiki="&wiki;"
    xmlns:rdf="http://www.w3.org/1999/02/22-
    rdf-syntax-ns#">
  <wiki:article rdf:about="&wiki;HashMap">
    <wiki:isA>
      <wiki:article rdf:about="&wiki;Map">
         <wiki:name>Map</wiki:name>
      </wiki:article>
    </wiki:isA>
    <wiki:name>HashMap</wiki:name>
  </wiki:article>
  <wiki:article rdf:about="&wiki;Tree">
    <wiki:canBe rdf:resource="&wiki;Map"/>
    <wiki:canBe>
      <wiki:article rdf:about="&wiki;Set">
        <wiki:name>Set</wiki:name>
      </wiki:article>
    </wiki:canBe>
    <wiki:name>Tree</wiki:name>
  </wiki:article>
  <wiki:article rdf:about="&wiki;ArrayList">
    <wiki:isA>
      <wiki:article rdf:about="&wiki;List">
        <wiki:name>List</wiki:name>
      </wiki:article>
    </wiki:isA>
    <wiki:name>ArrayList</wiki:name>
  </wiki:article>
</rdf:RDF>
```

The next listing shows the RDFS for the example above:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">
  <rdfs:Class rdf:ID="article">
    <rdfs:comment>A resource representing a
    wikipedia article</rdfs:comment>
  </rdfs:Class>
  <rdf:Property rdf:ID="name">
    <rdfs:comment>The name of an article
    </rdfs:comment>
    <rdfs:domain rdf:resource="#article"/>
    <rdfs:range rdf:resource="http://www.w3
    .org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>
  <rdf:Property rdf:ID="isA">
    <rdfs:domain rdf:resource="#article"/>
    <rdfs:range rdf:resource="#article"/>
  </rdf:Property>
  <rdf:Property rdf:ID="canBe">
    <rdfs:domain rdf:resource="#article"/>
    <rdfs:range rdf:resource="#article"/>
  </rdf:Property>
</rdf:RDF>
```

A considerable amount of the related work in Wikipedia is about extracting information from it in order to use Wikipedia as an ontological resource. Enriching the RDF schema with other kinds of information gathered through different analysis of Wikipedia could lead to an improvement of the proposed approach and to the overcoming of many problems of Semantic Web Ontologies.

The increase of interoperability is an advantage of the use of Wikipedia as an ontological resource, in fact as we showed with our methodology. It is possible to map content Wikipedia concepts. So the same concepts expressed in different documents can be expressed using the same URI. It comes by itself that our approach can also be used for Ontology matching. Suppose we have the need of describing relationships that are not covered by the semantic provided by Wikipedia. A solution derives from combining Wikipedia with domain ontologies for representing complex semantics. In all the scenarios we presented, annotations based on a common vocabulary like the one provided by Wikipedia could lead to interoperability between different ontologies or metadata formats.

4.2 Evaluation

The evaluation of automatic annotations is generally a very critical task for the reason that gold standards to compare annotations against are almost no-existent, furthermore annotations are subjective and the vocabulary that can be used to refer to the same object is very broad, so it becomes hard to discriminate the correct ones. Our solution to overcome the lack of gold standards is to use one provided by Mihalcea [31] for evaluating different disambiguation approaches.

The experiment task is about finding the correct sense of a word guessing the right title of a Wikipedia article. The gold standard consist of 112 manually semantic annotated Wikipedia articles. The annotators where also asked to choose words with a corresponded Wikipedia definition to describe the topics of the article. These keywords are used to avoid that inaccuracy in information extraction could influence the WSD task. Hence, they assumed that the keyword extraction stage produced 100% precision and recall.

Mihalcea in [31] used this gold standard for testing three WSD approaches. The first reported one is called knowledge based approach and takes into account the paragraph where the ambiguous word was found as a representation of the context (similar to LESK algorithm), and a second one called data-driven method imply a classifier and builds a feature vector with words in proximity of an ambiguous word found in the text and in the Wikipedia link of possible senses. A third approach combines these two. Our disambiguation process instead focuses only on the strong link analysis.

In the following table we report data about precision and recall for the three WSD approaches evaluated by Mihalcea in comparison with our disambiguation algorithm. Precision is calculated by counting the number or correctly annotated instances on the number of words covered by the system and recall is defined as number of correct annotations divided the total number of annotations that should have been generated by the system.

Table 1 Strong and Link based WSD Performance

Method	Precision	Recall
Knowledge-based	80.63	71.86
Feature-based learning	92.91	83.10
Combine(Knowledge+Feature)	94.33	70.51
Strong link-based	90.01	91.81

4.3 Comparison with Related Work

This section describes the differences between the novel contributions described in this chapter, and other related approaches we have described in section 3. Compared to the disambiguation approach of Mihalcea [30], the presented novel approach differs for the elements taken into account. When a word is found we analyze the whole content of the document and we extract its links and not just the paragraph where the ambiguous word was found.

Cucerzan in [7] describes an interesting approach for associating Name Entities in a corpus with Wikipedia definitions. The goal of this approach is similar to ours, the main difference is that we do not limit the corpus analysis to Name Entities and we considered also multilanguage material. They explored various strategies to decrease the numbers of attributes to consider. They reduce the context information by extracting entities with a certain number of mentions in the article or using some TF-IDF threshold. We also faced this problem in WSD. For learning about topic dependencies we considered only strong links among articles.

We also analyzed the possibility of using a Wikipedia knowledgebase such as DBpedia or Yago for replacing Wikipedia as a source of knowledge in our approach. We discarded this possibility because these knowledge bases contain only entities (such as person and places) and not abstract concepts as the ones we commonly have in the presentation material. In addition we do not know a priori with which property a term a can be searched, hence in our domain replacing Wikipidia free-text search would not be beneficial. On the other hand we are still looking a ibrid possibilities for replacing part of the computing in the disambiguation process with DBpedia.

5 Conclusions and Future Work

This chapter provides both a survey of the work performed different areas using Wikipedia as an ontological resource, and the description proof-of-concept application for automatically annotating the content of multimedia presentations, underlining the power of Wikipedia semantics. The approach has also been applied to a digital library project for organizing e-Learning material [13]. This result is achieved using a combination of Information Extraction, Natural Language Processing techniques and knowledge disambiguation and mapping against Wikipedia.

We are currently working on different aspects in order to improve the quality and usability of the solution. One aspect is about extending the annotations to Name Entities and creating a web service library so that it will be much easier for new applications to include the semantic annotations functionality. At the same time we are evaluating ways to remove the need of mantaning a local copy of Wikipedia for improving computational speed.

The advantages of having a broad domain coverage such as the one found in Wikipedia can reduce the shortcomings of Semantic Web efforts which tend to suffer from domain coverage, but also from issues of identity and reference. The problem of unique identifiers for resources is crucial for accomplishing semantic interoperability and efficient knowledge integration.

The use of the Wikipedia URI for identifying well-known concepts is one possible solution to the problem of identity mismatch and guarantees interoperability between domain ontologies, while the extensive ongoing research effort for extracting an ontological view from Wikipedia can provide a reference ontology for many domains.

References

1. Adafre, S.F., Jijkoun, V., de Rijke, M.: Fact discovery in wikipedia. In: WI 2007: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 177–183. IEEE Computer Society, Washington (2007), http://dx.doi.org/10.1109/WI.2007.57
2. Adafre, S.F., de Rijke, M.: Discovering missing links in wikipedia. In: LinkKDD 2005: Proceedings of the 3rd International Workshop on Link Discovery, pp. 90–97. ACM, New York (2005)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: ISWC/ASWC, pp. 722–735 (2007)
4. Auer, S., Lehmann, J.: What have innsbruck and leipzig in common? extracting semantics from wiki content. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 503–517. Springer, Heidelberg (2007)
5. Chernov, S., Iofciu, T., Nejdl, W., Zhou, X.: Extracting semantic relationships between wikipedia categories. In: 1st Workshop on Semantic Wikis (2006)
6. Ciaramita, M., Altun, Y.: Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 594–602. Association for Computational Linguistics, Sydney (2006), http://www.aclweb.org/anthology/W/W06/W06-1670
7. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP 2007: Empirical Methods in Natural Language Processing, Prague, Czech Republic, June 28-30, pp. 708–716 (2007), http://acl.ldc.upenn.edu/D/D07/D07-1074.pdf
8. Culotta, A., McCallum, A., Betz, J.: Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In: Proceedings of the main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pp. 296–303. Association for Computational Linguistics, Morristown (2006), http://dx.doi.org/10.3115/1220835.1220873
9. Ebersbach, A., Glaser, M., Heigl, R.: Wiki: Web Collaboration. Springer, Heidelberg (2005)
10. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2003)
11. Fields, K.: Ontologies, categories, folksonomies: an organised language of sound. Org. Sound 12(2), 101–111 (2007), http://dx.doi.org/10.1017/S135577180700177X
12. Fogarolli, A.: Word sense disambiguation based on wikipedia link structure. In: IEEE ICSC 2009 (2009)
13. Fogarolli, A., Ronchetti, M.: Intelligent mining and indexing of multi-language e-learning material. In: Tsihrintzis, G., et al. (eds.) 1st International Symposium on Intelligent Interactive Multimedia Systems and Services, KES IIMS 2008. SCI, vol. New Directions in Intelligent Interactive Multimedia, pp. 395–404. Springer, Heidelberg (2008)
14. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, Boston, MA (2006)
15. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 6–12 (2007)

16. Cui, G., Lu, Q., Li, W., Chen, Y.: Corpus exploitation from wikipedia for ontology construction. In: E.L.R.A (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
17. Giles, J.: Internet encyclopaedias go head to head. Nature 438(7070), 900–901 (2005)
18. Klein, G.O., Smith, B.: Concept systems and ontologies. Discussion between realist philosophers and ISO/CEN experts concerning the standards addressing ”concepts” and related terms (2005)
19. Gruber, T.: Tagontology - a way to agree on the semantics of tagging data (2005), http://tomgruber.org/writing/tagontology-tagcamp-talk.pdf
20. Hepp, M., Siorpaes, K., Bachlechner, D.: Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management. IEEE Internet Computing 11(5), 54–65 (2007), doi:10.1109/MIC.2007.110
21. Hu, M., Lim, E.P., Sun, A., Lauw, H.W., Vuong, B.Q.: Measuring article quality in wikipedia: models and evaluation. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 243–252. ACM, New York (2007), http://doi.acm.org/10.1145/1321440.1321476
22. Janik, M., Kochut, K.J.: Wikipedia in action: Ontological knowledge in text categorization. ICSC 0, 268–275 (2008), http://doi.ieeecomputersociety.org/10.1109/ICSC.2008.53
23. Atserias, J., Zaragoza, H., Ciaramita, M., Attardi, G.: Semantically annotated snapshot of the english wikipedia. In: E.L.R.A (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
24. Kamps, J., Koolen, M.: The importance of link evidence in wikipedia. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 270–282. Springer, Heidelberg (2008)
25. Krizhanovsky, A.: Synonym search in wikipedia: Synarcher. arxiv.org http://arxiv.org/abs/cs/0606097v1; Search for synomyms in Wikipedia using hyperlinks and categories
26. Lankes, R.D., Silverstein, J., Nicholson, S., Marshall, T.: Participatory networks the library as conversation. Information Research 12(4) (2007), http://iis.syr.edu/projects/PNOpen/Particiaptory Networks.pdf
27. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: SIGDOC 1986: Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26. ACM, New York (1986), http://doi.acm.org/10.1145/318723.318728
28. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 266–272. ACM, New York (2004), http://doi.acm.org/10.1145/1008992.1009039
29. Ramos, M.A., Rambow, O., Wanner, L.: Using semantically annotated corpora to build collocation resources. In: E.L.R.A (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
30. Mihalcea, R.: Using wikipedia for automatic word sense disambiguation. In: Proceedings of NAACL HLT 2007, pp. 196–203 (2007), http://www.cs.unt.edu/~rada/papers/mihalcea.naacl07.pdf

31. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 233–242. ACM, New York (2007)
32. Milne, D.: Computing semantic relatedness using wikipedia link structure. In: New Zealand Computer Science Research Student Conference (2007)
33. Milne, D., Medelyan, O., Witten, I.H.: Mining domain-specific thesauri from wikipedia: A case study. In: WI 2006: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 442–448. IEEE Computer Society, Washington (2006), http://dx.doi.org/10.1109/WI.2006.119
34. Nguyen, D.P.T., Matsuo, Y., Ishizuka, M.: Relation extraction from wikipedia using subtree mining. In: AAAI, pp. 1414–1420. AAAI Press, Menlo Park (2007)
35. Noruzi, A.: Folksonomies (un)controlled vocabulary? Knowledge Organization 33(4), 199–203 (2006), http://noruzi.blogspot.com/2007/07/folksonomies-uncontrolled-vocabulary.html
36. Obrst, L.: Ontologies for semantically interoperable systems. In: CIKM 2003: Proceedings of the Twelfth International Conference on Information and Knowledge Management, pp. 366–369. ACM Press, New York (2003), http://doi.acm.org/10.1145/956863.956932
37. Ollivier, Y., Senellart, P.: Finding related pages using Green measures: An illustration with Wikipedia. In: Proc. AAAI, Vancouver, Canada, pp. 1427–1433 (2007)
38. Pask, G.: Conversation, cognition and learning: A cybernetic theory and methodology. Elsevier, Amsterdam (1975), http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0444411933
39. Ponzetto, S.: Creating a knowledge base from a collaboratively generated encyclopedia. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Doctoral Consortium, Rochester, N.Y., pp. 9–12 (2007)
40. Ponzetto, S., Strube, M.: Deriving a large scale taxonomy from wikipedia. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007), Vancouver, B.C., pp. 1440–1447 (2007)
41. Roth, M., im Walde, S.S.: Corpus co-occurrence, dictionary and wikipedia entries as resources for semantic relatedness information. In: E.L.R.A (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
42. Ruiz-Casado, M., Alfonseca, E., Castells, P.: From wikipedia to semantic relationships: a semi-automated annotation approach. In: SemWiki (2006)
43. Schaffert, S.: Ikewiki: A semantic wiki for collaborative knowledge management. In: 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2006, pp. 388–396 (2006)
44. Schonhofen, P.: Identifying document topics using the wikipedia category network. In: WI 2006: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 456–462. IEEE Computer Society, Washington (2006)
45. Siorpaes, K., Hepp, M.: Ontogame: Weaving the semantic web by online games. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 751–766. Springer, Heidelberg (2008)
46. Snoek, C., Worring, M.: Multimodal video indexing: A review of the state-of-the-art. In: Multimedia Tools and Applications, vol. 25, pp. 5–35 (2005)

47. Strube, M., Ponzetto, S.: WikiRelate! Computing semantic relatedness using Wikipedia. In: Proceedings of the 21st National Conference on ArtificialIntelligence (AAAI 2006), Boston, Mass., pp. 1419–1424 (2006)
48. Suchanek, F., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Research Report MPI-I-2007-5-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany (2007)
49. Suh, S., Halpin, H., Klein, E.: Extracting common sense knowledge from wikipedia. In: Proc. of the ISWC 2006 Workshop on Web Content Mining with Human Language technology (2006), http://orestes.ii.uam.es/workshop/22.pdf
50. Syed, Z., Finin, T., Joshi, A.: Wikipedia as an ontology for describing documents. In: Proceedings of the Second International Conference on Weblogs and Social Media. AAAI Press, Menlo Park (2008)
51. Thomas, C., Sheth, A.P.: Semantic convergence of wikipedia articles. In: WI 2007: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 600–606. IEEE Computer Society, Washington (2007), http://dx.doi.org/10.1109/WI.2007.93
52. Twidale, B.S.M.B.: Assessing information quality of a community-based encyclopedia. In: Proceedings of the International Conference on Information Quality, pp. 442–454 (2005)
53. Uren, V.S., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. J. Web Sem. 4(1), 14–28 (2006)
54. Vercoustre, A.M., Thom, J.A., Pehcevski, J.: Entity ranking in wikipedia. In: SAC 2008: Proceedings of the 2008 ACM Symposium on Applied computing, pp. 1101–1106. ACM, New York (2008), http://doi.acm.org/10.1145/1363686.1363943
55. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic wikipedia. In: Proceedings of the 15th International Conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26 (2006), http://www.aifb.uni-karlsruhe.de/WBS/hha/papers/SemanticWikipedia.pdf
56. Voss, J.: Measuring wikipedia. In: Proceedings International Conference of the International Society for Scientometrics and Informetrics: 10 th (2005), http://eprints.rclis.org/archive/00003610/
57. Voss, J.: Collaborative thesaurus tagging the wikipedia way (2006), http://arxiv.org/abs/cs.IR/0604036
58. Wang, G., Yu, Y., Zhu, H.: Pore: Positive-only relation extraction from wikipedia text. In: Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 575–588. Springer, Heidelberg (2007), http://iswc2007.semanticweb.org/papers/575.pdf
59. Wang, J.Z., Boujemaa, N., Bimbo, A.D., Geman, D., Hauptmann, A.G., Tesić, J.: Diversity in multimedia information retrieval research. In: MIR 2006: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 5–12. ACM, New York (2006), http://doi.acm.org/10.1145/1178677.1178681
60. Wang, J.Z., Boujemaa, N., Chen, Y.: High diversity transforms multimedia information retrieval into a cross-cutting field: report on the 8th workshop on multimedia information retrieval. SIGMOD Rec. 36(1), 57–59 (2007), http://doi.acm.org/10.1145/1276301.1276315

61. Watanabe, Y., Asahara, M., Matsumoto, Y.: A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 649–657. Association for Computational Linguistics, Prague (2007), http://www.aclweb.org/anthology/D/D07/D07-1068
62. Weber, N., Buitelaar, P.: Web-based ontology learning with isolde. In: Proc. of ISWC 2006 Workshop on Web Content Mining with Human Language Technologies (2006), http://orestes.ii.uam.es/workshop/4.pdf
63. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 41–50. ACM, New York (2007), http://portal.acm.org/citation.cfm?id=1321440.1321449, doi:10.1145/1321440.1321449
64. Yu, J., Thom, J.A., Tam, A.: Ontology evaluation using wikipedia categories for browsing. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 223–232. ACM, New York (2007), http://doi.acm.org/10.1145/1321440.1321474
65. Zesch, T., Gurevych, I.: Analysis of the wikipedia category graph for nlp applications. In: Proc. of the TextGraphs-2 Workshop (2007), http://acl.ldc.upenn.edu/W/W07/W07-0201.pdf
66. Zesch, T., Gurevych, I., Mühlhäuser, M.: Analyzing and accessing wikipedia as a lexical semantic resource. In: Biannual Conference of the Society for Computational Linguistics and Language Technology (2007)

Toward a Methodology of Collaborative Modeling and Simulation of Complex Systems

Benoit Gaudou, Nicolas Marilleau, and Tuong Vinh Ho

Abstract. Complex Systems Modeling and simulation are activities where collaboration between researchers with complementary points of view, skills and knowledge is well established and constitutes the usual working way rather than an exception. To manage difficulties of traditional modeling projects (how to translate informal knowledge into an implemented software, which intermediate model needed to be created...), lots of methodologies have been proposed. But to be well adapted to the particular case of complex systems, they need to go further. In this paper, we thus aim at providing a complete methodology taking into account the collaborative aspect of the Complex Systems Modeling and Simulation task. We thus describe collaboration between various participants and propose a minimal set of tools necessary for a modeling platform.

Keywords: Collaborative modeling, agent-based simulation, distributed systems.

1 Introduction

Modeling and Simulation (M&S) of Complex Systems (CS) are activities where collaboration between researchers is well established and constitutes the usual

Benoit Gaudou
UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique (IFI),
Ha Noi, Viet Nam
e-mail: benoit.gaudou@alumni.enseeiht.fr

Nicolas Marilleau
UMI 209 UMMISCO, Institut de Recherche pour le développment (IRD), Bondy, France
e-mail: nicolas.marilleau@ird.fr

Tuong Vinh Ho
UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique (IFI),
Ha Noi, Viet Nam
e-mail: ho.tuong.vinh@auf.org

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 27–53.
springerlink.com

working way rather than an exception. Indeed studied systems are so complex that researchers from various domains with complementary points of view, skills and knowledge are requested to an accurate study of the systems. The old-fashioned image of the researcher able to collect data, conceptualize and model informal description of the systems, implement a simulator and analyze results all by itself should be replaced by the one of an interdisciplinary group of researchers typically including thematicians (*i.e.* experts in the studied domains), mathematicians and computer scientists. For such a group, the model (and the simulation) are: (i) the prime motivation of the group formation, (ii) the collective goal and (iii) the object on which they work. The model is indeed the *alpha* and the *omega* of these researchers' collective work.

In addition to difficulties induced by any research work in general and modeling and simulation projects in particular (such as how to go from informal knowledge to a software, which intermediate models to create, how to create efficiently these models and so on), geographical separations, cultural differences or language problems create new difficulties and make international and interdisciplinary collaborations less efficient than local interactions.

A lot of work have thus analyzed the creation process during the M&S task and exhibited methodologies to improve it. First of all let begin by defining what we call a modeling methodology. Following [28], a methodology is composed of two linked elements:

- a *method*, *i.e.* a reproducible process driving the modeler by fixing intermediate subgoals to achieve;
- a set of *tools*: conceptual ones (*e.g.* languages or graphical representation formalisms), but also softwares helping to apply this process and to achieve each subgoal.

It is important to highlight that a methodology cannot be limited to only one element and should contain both the method and the tools.

Despite the high importance of the collaboration in the M&S process due to the need for researchers with complementary skills to build a good model, few methodologies take this issue into account. As far as we are aware, no one describes in details the roles of various stakeholders in the *method*, nor provides *tools* supporting this collaboration. In particular, all these issues naturally impose the use of tools to efficiently mediate interactions and to improve collaborative work between teammates. The advents of information technologies and in particular of large-scale networks since two decades have enabled the development of a plethora of tools allowing collaboration, from e-learning to industrial or scientific collaborative works including documents sharing, communications and development tools.

In this paper, we thus aim at providing a complete methodology taking into account the collaborative aspect of the M&S task. In particular we view it as a joint task which thus needs a specific method and dedicated tools to take into account this particular feature. Another important feature that we want to integrate in our methodology is the deep link existing between the modeling and the simulation processes: the model is not written in the stone when its implementation begins, on the contrary the simulation provides precious information about the model value, and the analysis of its results imposes modifications needed to improve the model. Thus modeling, on the one hand, and simulation, in the other hand, of complex systems cannot be two independent processes with separated tools. On the contrary interactions between all these tools are necessary. Moreover to be able to characterize all the steps of the modeling process we have to choose a particular paradigm in which the model will be developed. In our case we choose to limit our methodology to agent-based modeling and simulation.

This chapter is structured as follows. Section 2 presents an example of a M&S project in the domain of ecology. This example highlights the difficulties of such a work and the improvements that collaborative tools could bring. We thus present the proposed methodology in two parts. We begin in Section 3 by proposing the method and we provide a specification of the necessary tools in Section 4. Finally Section 5 ends the chapter with a discussion about the proposed methodology by comparing it with existing collaborative modeling tools.

2 Example of a Real Modeling Problem: The Case of Sworm

2.1 Presentation of the Example

The aim of the Microbes project is to identify soil functioning by studying soil biota (microorganisms, fauna and roots) evolution. In this context, a work (presented in [7, 22]) aims at reproducing the earthworms influence on the soil structure and the nutrient availability by simulation. For this purpose an agent-based model has been developed: Sworm.

The Microbes project provides a bottom-up approach based on multi-agent systems, which gives the opportunity to researchers to describe the system at a micro level (*e.g.* earthworms and their local soil environment) in order to observe, during the simulation, macroscopic changes (*e.g.* soil structure evolution, organic matter dynamics and microbial functions).

The soil structure can be defined as the spatial organization of solid mineral, organic particles and pore space. It is of great importance for soil functioning because it drives ecosystem services (carbon sequestration, emission of greenhouse gases, nutrient cycling, primary productivity...). The structure results from biotic and abiotic factors. Among biotic factors, numerous studies have shown the importance of organic matter, micro-organisms, roots and invertebrates. Earthworms are known to play a key role in the soil structure formation and maintenance through a continuous production of biogenic structures (casts and burrows).

Modeling this complex system is faced to the difficulty to describe the physical soil environment because a soil is a multi-scale heterogeneous, three-dimensional and dynamic environment. An approach based on the fractal theory (often used in soil sciences [6]) was chosen to model such a real complex environment; it was integrated into a Multi-Agent System (MAS). The MAS allows to simulate situated agents (*e.g.* earthworms) in a virtual world (*e.g.* soil).

Such as every interdisciplinary project, Microbes provides important results (i) for soil sciences by opening a new research way and giving results that could not be obtained before by experiments, but also (ii) for the multi-agent domain by providing a dynamic structure of environment (APSF-Agent Porus Solid Fractal) that permits to model a complex multi-scale world in a simplified but not simplistic way.

Sworm model, simulation results and the APSF structure are the results of an efficient collaboration between biologists and computer scientists. In such a modeling project, collaboration between actors (coming from various research domains) is the key to success.

2.2 Modeling and Simulation Process

From a real case study (the Lamto soil in ivory coast), a typical modeling-simulation process was followed, step by step, by the Microbes group composed of biologists and computer scientists in order to develop the Sworm model and simulator. Project members created successively: (i) a domain model; (ii) a conceptual UML model; (iii) a simulator; and (iv) experiment results.

The Microbes project did not progress in a linear way. Established models and the simulator were often discussed and refined. For example, the execution of the first version of the simulator showed lacks in the domain model and in the conceptual model. Improvements were added to the models and to the simulator. Then new results were obtained that permitted to identify new necessary improvements and so on. The models and the simulator result of an iterative and incremental modeling process in which a well collaboration between scientists was the key to success.

At the beginning, the Microbes project was punctuated by several exchange difficulties between computer scientists and biologists because of:

- *the project goal which was not clearly identified*. Biologists of the Microbes group were inexperienced in the field of modeling. For this reason, they did not know possibilities given by modeling technics in soil sciences. In addition, biologists and computer scientists had different scientific objectives. Participants of the project had to negotiate a long time to determine with accuracy the aim of the project.
- *misunderstandings*. Each community uses its own scientific language. In addition a different meaning can be given in computer sciences and biology for a same concept.
- *the forgetting of knowledge*. Due to the lack of tools to save exchanges, many and many knowledge were lost between two meetings.
- *the distance between project participants*. Members of the projet are not in the same research unit, they are 600 kilometers far. They could not organize regular meetings. Consequently, the progress of the project sped down and misunderstanding appeared.
- *a lack of dedicated tools*. There were not dedicated tools that support and save exchanges between biologists and computer scientists. Ad-hoc softwares were

used such as spreadsheets or diagram designers that did not ensure history, interoperability and so on.

To provide a solution for these problems, many and many meetings were organized (one week per month) in which one biologist and one computer scientist worked together in the same place. Major initial problems have decreased with the exchanges and the evolution of the modeling process. The frequency of these meetings were also the key to success of the project. It permitted to create a strong link between two persons coming from two scientific communities and to engage very interesting exchanges between them. During the Microbes project these two researchers had (i) learnt to talk together by using a scientific language which was at the middle of soil science and computer science and (ii) developed tools (diagrams, formalisms, and so on) to support exchanges.

In this context, the use of a groupware should have been an interesting way to lead, formalize and save the modeling-simulation process and exchanges. This framework may have reduced the distance between Microbes members, facilitated and sped up the modeling process and permitted to construct a base of knowledge resulting from exchanges, for this projet and other ones.

3 Modeling Method

Numerous methodologies have been developed to help scientists to construct their models and implement simulators. They try to drive modelers from the first step of the modeling process to the end of the implementation, which should also induce verification and validation steps *via* simulations. An overview has been presented in [21]. Note that in the sequel we will include simulation in the modeling process: in this paper the simulation is considered as a (and perhaps the unique) way to test and validate the model. Simulation as a decision support tool is out of the scope of this paper. As detailed in Section 1, a whole methodology should be composed of a *method* and *tools* supporting this method. We present the method in this section and tools in the following one.

3.1 Examples of Modeling Methodologies

Lot of works have analyzed the creation process in M&S task and exhibited methodologies to improve this complex work. We can cite for example the well-known and generalist GAIA [36]. It is said to be generalist [21] because it is a software engineering method, *i.e.* it can be used to develop any software (simulators or others...). Among other generalist methodologies we can cite TROPOS [8, 34] that drives users from the first description of the system to the end of the implementation. In addition to these methodologies, the AUML language [4] has been developed as a modeling language that extends the UML language to represent agents and their interactions.

These methodologies are very interesting and useful to develop MAS or agent-based simulators. Nevertheless we can object that they are mainly centered on tasks to perform and models to produce. Indeed they are not interested to various people designing the model and to their particular skills. In addition, they are not understandable by scientists who have not a solid background in agent-based modeling. Finally they are often linear: lot of approaches disconnect the modeling process from the simulation, which is necessary to validate models or bring new ideas to every people. We thus advocate a methodology with very strong links between the modeling process, simulation and tests.

In particular we argue that the modeling process should be close to agile programming paradigms, such as eXtreme Programming (XP) [5]. Note that in XP the development cycle of the model is quite short with frequent feedbacks from customers on the product. Customers take part to the development process from the beginning to the end in order to give quick feedbacks to the products; the aim is to have a product the closest possible to what they want (and not to what development team has understood they wanted). Similarly we aim at a methodology where thematicians can take part to most steps of the process to give their own stand point, ideas and desiderata. They can evaluate models proposed. To formalize such a process we need to introduce explicitly the various actors of the process. To this purpose we take as start point the role-based methodology proposed by Drogoul *et al.*in [10].

3.2 Presentation of Drogoul et al.*'s Methodology [10]*

In [10], authors argue that existing methodologies, such as Fishwick's [13] or Gilbert and Troitzch's [15] ones, suffer from important drawbacks: (i) they do not take into account specificities of agent-based simulations and could be applied to any kind of simulations, (ii) they are task-oriented rather than model-oriented which could permit to highlight features and specific difficulties of the model traduction.

To round these drawbacks, Drogoul *et al.* propose a methodology based on the notion of role: the modeling and simulation process (in particular in the case of agent-based M&S of Complex Systems) involves several people, with particular skills, points of view, goals... Every stakeholder will thus not have the same role in the process and will not perform same tasks.

Figure 1 summarizes Drogoul *et al.*'s methodology. Authors distinguish three roles. The domain expert, also named **Thematician**, defines the aim of the modeling and simulation project (*i.e.* he defines the initial question which should be answered by the simulation) and provides knowledge and data about the system. In particular, he brings two sets of knowledge: macro-knowledge (*macro-K*),*i.e.* knowledge from observation of the global target system, and micro-knowledge (*micro-K*), *i.e.* knowledge from observation and assumptions of the individual components of the target system. The thematician's motivations to undertake an agent-based modeling and simulation project is often the will to understand the link between the micro and the macro level. The thematician uses micro-K to build an often informal *domain model* that describes real agents (agents observable in the target system), their parameters and behaviors.

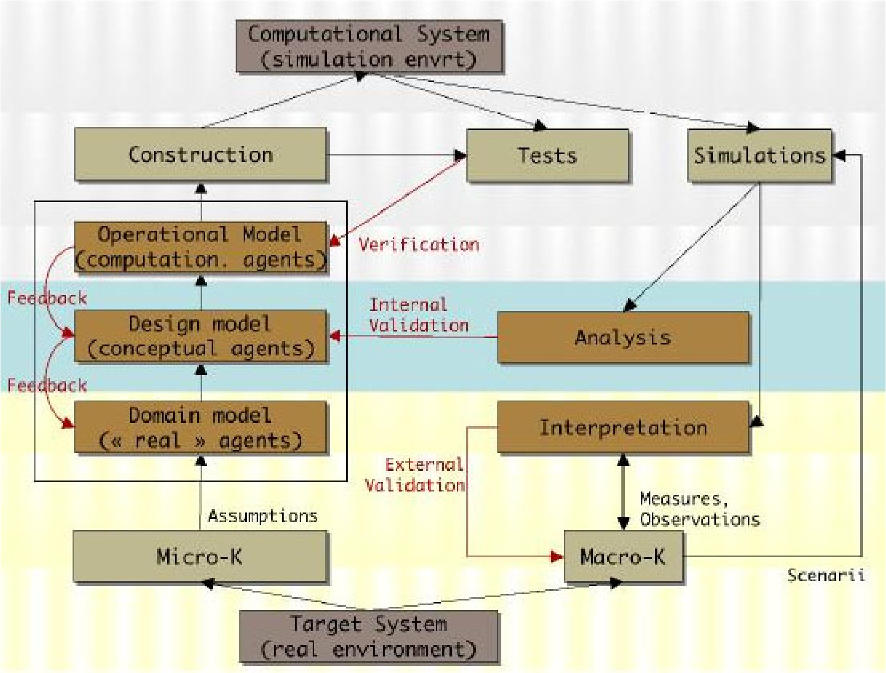


Fig. 1 Drogoul *et al.*'s cycle [10]

The implementation of the design model is not possible directly because it is often informal, ambiguous and it lacks lot of details needed and contains lot of details superfluous for the implementation. Thus the **Modeler** aims at translating this first model into an unambiguous *design model*; this model should be described in terms of MAS concepts, such as agents, environment, communication and so on.

From this design model, the **Computer Scientist** should propose an *operational model* before its implementation. This last model allows both the modeler and the thematician to understand what has been implemented and to be able to comment or to modify it if needed. The *construction* of the *computational system* follows the creation of this model.

The development of these three models allows all the stakeholders to better understand how elements of the domain model impact the simulation. Note that the production of each model induces *feedbacks* on lower levels models, the creation of a model is thus the result of an iterative process rather than a one-shot task. The simulator resulting from this process should then be verified and validated. The computer scientist should drive *tests* to verify the simulator relatively to the operational model. *Simulations* lead to validation with the conceptual model and with the macro-K defined by the thematician.

3.3 How to Anchor Collaboration in This Methodology?

Drogoul *et al.*'s work [10] described above presents two main advantages for us: (i) it is a complete attempt of conceptual methodology from the real target system toward the computational system, taking into account feedbacks from the simulation to improve models, and (ii) without tackling explicitly the collaboration problem, it takes into account various roles of people involved in the M&S project, which is the first step to consider the collaboration.

We want to go through this approach. Authors have detailed each stakeholder's role and responsibilities. But interactions between various stakeholders have not been investigated: they seem to have independent tasks. Actually interactions between various stakeholders are very important during the modeling process (and also during simulation stage as we will highlight it below). We detail in the sequel interactions and thus collaborations that exist between these individuals and what are the support of this collaboration. This analysis, in particular about the collaboration support, is necessary to design useful tools to support the collaborative aspect of the modeling process.

3.3.1 Collaboration in the Elaboration of the Domain Model

Drogoul *et al.*'s methodology assigns the responsibility of the domain model to the domain expert. We argue, supported by experiences in interdisciplinary modeling and simulation projects such as the Microbes one, that it is much more efficient that the thematician produces this model collaboratively with the modeler rather than completing it alone and transmitting it afterwards to the modeler who will use it to do the conceptual model. Indeed, as highlighted by the Microbes project, this first step allows thematician and modeler to develop a common language. Without a close collaboration, it could be very time consuming to reach a common understanding of concepts and misunderstandings can run all along the project. Another very important point is the fact that thematicians have often only a little idea of what is possible to do from the computational point of view, what is not, what is hard, which information is needed and which is superfluous. The modeler has a better general view and shall thus drive the thematician. He has thus also a kind of knowledge extraction support task. In particular it is usual that the thematician himself does not know exactly what he wants to do with the simulation; the question to which the model should answer is not so clear. It is also the role of the modeler to point out these issues to thematicians who often had never worked on a modeling and simulation project.

To summary we argue that, for the stage of creation of the domain model, collaboration between the thematician and the modeler is required. Moreover we argue that to be allowed to express freely their idea with all the details permits a better mutual understanding: words in their context should be easier to understand.

3.3.2 Collaboration in the Elaboration of the Design Model

From this description of the problem, the modeler should produce the conceptual model. We argue that modeler should begin by extracting concepts from the textual description, properties of these concepts and links between them... A suitable representation would be under the form of an ontology. The creation of this ontology gives also the opportunity to every participants to clarify all concepts of the studied system. Thus it implies naturally the collaboration between modeler and thematician.

From this ontology, the modeler extracts the entities, that will become the agents of the future MAS, and their parameters. He gives a clear description of each agent and each parameter. In particular, for each parameter he has to define its type, its value range, its unit... The need of some constant values and thus of data characterizing each kind of agent can also appear.

3.3.3 Collaboration in the Elaboration of the Operational Model

The modeler will then design the agents' behavior from both domain and design models. By experience, it appears that it is more efficient to involve the thematician in this step. This induces the development of a representation language intuitive enough to be understood by any scientist. A simple graphical language should thus be used. There are two main benefits to involve thematicians in this process. On the one hand they can check quickly designed behaviors. On the other hand this gives them a better understanding of how the simulation will work and thus will improve the simulation and result analysis steps.

By designing agents' behaviors, some lacks in the previous descriptions will appear. One of the most often lack concerns the data. For example, in the Microbes projet, a lack of data has appeared to model earthworms by agents. In soil sciences, no research has tried to qualify with accuracy earthworm behavior. This issue is also one for the thematician because the collect of data is time (and money) consuming. Thus it is important to detect quickly lacking data in order not to be blocked in the project. This also imposes efficient collaboration and communication between stakeholders.

We do not tackle the problem of the whole operational model in this paper because, as it will be shown in the sequel, we will use a generic platform such as NetLogo [35] or GAMA [2] to create the MAS simulator. Actually we only have to write files (in a specific language) describing agents, their parameters and their behaviors. The platform manages everything else. The operational model can thus be limited to the description of behaviors in this case. We do not detail the debugging of the behavior implementation, we let it to the modeler/computer scientist.

3.3.4 Collaboration in the Validation of Models

The validation of the model thanks to thematician's Macro-K implies to launch simulations and observe their results. This step cannot be done without both modeler and thematician. The former knows deeply the simulator (computational system) whereas the latter knows the target system. Once more they have to collaborate because they have complementary skills and knowledge. This step needs thus tools to collaboratively run simulations. Note that simulations can show data lacks in micro- or macro- knowledge sets or will induce big model revisions. The collaboration between both parts is thus necessary to determine which should revise depending on simulation results.

The definition of a complete methodology also implies to provide some tools able to support its conceptual aspects. In our case we focus on providing tools allowing collaborative work during the modeling process and will use a collaborative simulation tool that we are developing [25] for collaborative simulation. We first investigate available tools for agent-based modeling.

4 Modeling Tools

4.1 What about Existing Agent-Based Modeling Tools?

There exist lot of tools helping users to create their models. In the sequel, we describe some of them and highlight their limitations in particular in terms of collaboration.

4.1.1 Pedagogical Tools

The general purpose of such softwares is to allow people without programming skills to create multi-agent simulations by graphically creating agents' behaviors. Among the multitude of applications we can cite Etoys [17], Stagecast Creator [31], AgentSheets [29] or Scratch [20].

The user often begins by placing or drawing (for Etoys) agent sprites in an environment. He can graphically define for each agent their variables (aiming at representing agents' internal states) and behaviors. Behaviors are often represented by a script based on an if-then-else structure: if such condition is fulfilled then the agent executes such actions or updates such variables... Conditions can be described in terms of agents' localization, physical features (such as the intersection between drawn parts in Etoys), appearance or a interaction with the user (mouse clicked or key pressed...). Actions to be executed include movement, appearance changes, noises, variables changes or operations on variables...

Thus it appears that such modeling tools are very user-friendly and easy to learn. They are moreover clearly children-oriented. Thanks to such features kids are able to create quickly some running simulations. They are thus very interesting tools to discover programming of robots (Etoys) and/or multi-agent modeling and simulation. Nevertheless they are still too limited for a "serious" use. In particular, the characterization of the environment remains very basic and behavior languages are too limited for actual simulations. Moreover (except Agentsheets), they seem to be dedicated to small simulations with only few agents. We can remark that Scratch has the most evolved graphical language, which has for a big part influenced another very interesting tool, StarLogo TNG, presented below.

4.1.2 StarLogo TNG

StarLogo TNG [18] is a very simple and user-friendly StarLogo-based application. It is composed of a very beautiful and intuitive Graphical User Interface (GUI) inspired by the Scratch one, but allowing to use all the power of StarLogo language, coupling with a view of the environment and of the running simulation. By its use of specific shapes, color code and explicit lists of available functions, it can really be viewed as an IDE for StarLogo aiming at facilitating programming.

Actually StarLogo TNG is no more than a graphical interface to implement StarLogo models. And from this point come its pros and drawbacks. It is very easy for people having already used StarLogo or NetLogo to adapt to StarLogo TNG. But this graphical programming becomes quickly heavy compared to classical free textual programming. Modelers will thus feel freer and will program faster with NetLogo than with StarLogo TNG. Moreover it is not clear that it could be well adapted for a use by thematicians. Although the graphical interface can appear easier to use for non-computer scientists, StarLogo TNG saves all the expressivity of the StarLogo language and with it all its complexity. A thematician without any programming skills could thus have lot of difficulties to be able to implement alone his model.

4.1.3 Repast Symphony

Repast J is a well-known open source framework to build agent-based simulator platforms. It is widely used in multi-agent platform projects such as GAMA [2]. Its new released, called Repast Symphony [27], has adopted a graphical interface to define agents behaviors. Nevertheless the graphical language remains quite poor: indeed it includes only blocks of tasks, conditionals, and loops. In a task block, some instructions in Groovy language have to be hand-implemented. A wizard has been developed to help writing such instructions blocks but in this first attempt it remains hard to use.

This graphical interface is unadapted for users able to implement in Groovy/Java and avoid them to use programming help tools such as auto-completion. Nevertheless it needs some programming skills and a quite good understanding of the logic behind programming to be used and thus is unadapted for thematicians.

Despite the above critics, the use of a similar simple language can have an interest for our purpose. Although we argue that it is not well adapted for the programming of the model, it could be useful for the representation of the algorithms. Each task block can be viewed as a microbehavior. The interest is that each microbehavior can have a intuitive meaning for a non-computer scientist. The decomposition of a behavior in microbehaviors logically linked by, among others, conditionals and loops can allow interesting discussions between each part of the project on algorithms. The inner coding task could be dedicated to computer scientist with specific tools.

4.1.4 Modeling4All

Modelling4All [16] is a web-portal allowing users to model and simulate online. It is based on NetLogo.

The key idea of Modeling4All is that agents' behavior can be decomposed in terms of micro-behaviors. All the micro-behaviors are executed without any order or link. They are small, coherent and independent program fragments. Modeling4All also proposes to users a library of micro-behaviors implemented in NetLogo. The library has been created to include micro behaviors for 2 models: Disease Models and sugarscape Models. Each micro-behavior is associated with a wiki page describing it in detail. It also allows users to edit online these behaviors. Users can also upload their own micro-behaviors.

Practically a new user creates an online project. He creates agents and gives them some micro-behaviors from the library. He can then choose to execute an online simulation or to export his model as a NetLogo file. Note that an history of every modifications is recorded.

The micro-behavior library provided by this web-portal is very interesting. But the simultaneity of the execution of micro-behavior appears quite counterintuitive. In combination with Repast Symphony features presented above, this could give a very interesting approach of the modeling process.

4.1.5 Mimosa - Cormas

The tool Mimosa [23, 24] has been created to support a particular methodology that can be decomposed in five steps:

- the conceptual model step, that corresponds to the phase of definition of the ontology (*i.e.* definitions of categories, attributes and relationships);
- the dynamics modeling step: choice of the paradigm and implementation of behaviors;
- the concrete modeling step: initialization of the model
- the simulation specification step: which attributes will be fixed, which ones will be manipulated by the user, what are the outputs?
- the simulation step.

This application appears quite complete but it does not take into account the collaboration problem.

Albeit all these softwares propose modeling tools, no one takes into account the collaborative nature of this process (Modeling4All is the lone one that introduces light collaborative features). We thus need to develop our own application to support and help the modeling methodology presented above.

4.2 Set of Tools for a Modeling and Simulation Platform

During the modeling process we have distinguished several steps; for each a particular task has to be realized (description of the system, conceptual model and so on).

Each task has its own specificities, language or formalism; dedicated tools should thus be developed for each one. In consequence a modeling and simulation platform should offer, in addition to general synchronous and asynchronous tools for the communication, specific tools to improve collaboration in these tasks:

- Definition of the problem to be modeled;
- Description of the concept ontology of the system (which entities? Which links between these entities?);
- Definition of various agents with their properties (states variables) and parameters;
- Design of behaviors;
- Instantiation of a simulation.

Coherence maintenance between models

By building several representations of the same system, there is a high probability that thematicians, modelers and computer scientists will introduce incoherence between models. It could be interesting to give some coherence maintenance (between models) features to our tools in order to help stakeholders to improve the global model. We do not want to have a very rigorous and strict synchronization system; we only want that, depending on what they have created in a model, the tools give some advises to improve other models and thus maintain a kind of coherence between various models.

Figure 2 associates to each stage of the Drogoul *et al.*'s methodology the tasks to be done, which induce the development of dedicated tools. In the sequel we detail each task and associate tools needed to build models; we also detail how to ensure coherence between each of these models. Table 1 provides a summary of the discussion.

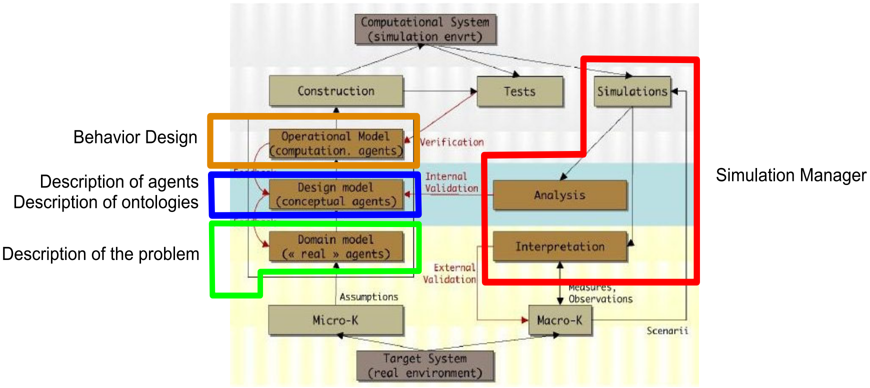


Fig. 2 Needed functionalities for a collaborative Modeling platform

Table 1 Summary

	Synchronous tools	Asynchronous tools
General	- Textual/audio chat - Visioconference - Whiteboard	- Wiki - Forum - RSS stream
Step 1: Domain model	- Textual editor	- Commentary on the text - File deposit and sharing
Step 2: Ontological model	- Graphical editor * mind-mapping mode * free mode	- Commentary on the graphics
Step 3: Design model	- Creation/deletion of agents - Add/remove of agents' parameters - Add/remove global parameters - Values of each field	- Commentaries on each parameter - Request on each parameter - TODO/TOFIND list (data)
Step 4: Operational model	- Graphical editor of the program flow - Microbehavior library - Graphical programming	- Graphical highlighting of TODO tasks - Commentaries of the graphics
Step 5: Simulation	- Initialization pane - Simulation control pane	

4.2.1 Inventory of Collaboration Tools Needed

It is important to note that the most important tools to allow collaboration between stakeholders are communication tools. Communication is essential at each step to coordinate efforts or simply to exchange ideas. Following [11], we make a distinction between synchronous (when stakeholders are connected at the application and work on models at the same time) and asynchronous (when one stakeholder works on models whereas others are offline) collaboration. Each of these two kinds of collaboration requires specific tools from the communication point of view.

Synchronous collaboration

It appears that synchronous collaboration, when it is possible, is the most efficient way to collaborate because it allows people to directly discuss, exchange ideas and react to others' actions... Thus tools should allow stakeholders to freely communicate by audio, video and/or written chats.

Beside generic communication tools, it would be interesting to add to tools a kind of *awareness* feature that allows users to know what others are doing or which model or parameter they are modifying. This feature would avoid many useless and time-consuming interactions between users to coordinate their work. We should then add a panel showing in which model users are working and, in each model view, what they are doing by highlighting on which element they are focused. This focus highlight can also be used by a user to show something to others. These features help users to quickly localize what a user is talking about and thus help mutual understanding between stakeholders.

Following tools are thus needed in all models to take the best advantage of synchronous work:

- Synchronous communication:
  - textual and/or audio chat, visioconference to communicate freely;
  - whiteboard to draw freely on new or existing pictures, synchronously with other partners.
- Awareness:
  - panel listing on which model users are working;
  - highlights showings elements focused.

Asynchronous collaboration

Asynchronous collaboration has been shown to be much less effective than synchronous one, but it remains essential in a collaborative project especially when partners are geographically separated and unable to work synchronously due to time difference. Moreover an asynchronous work gives time to think and provide deeper solutions to problems and thus are also needed when hard issues appear. So asynchronous tools are needed to put commentaries on others work, to post developed responses to particularly hard problem or to react to important decisions.

Following tools are thus needed in all models to take the best advantage of asynchronous work:

- Asynchronous communication:
  - collaborative communication tools (including wiki, forum, RSS stream for news on the forum...);
  - possibilities to post commentaries on each element of the model;
  - logs of textual chat should be kept on the server.
- A tool showing what has been modified since the user's last connection;
- A collaborative TODO list: any user can post things to do. These tasks can be associated to all users, to groups of users (*e.g.* thematicians) or to a particular user. In particular, it will be the place for data requests.

After having described tools needed for the whole project, we detail in the sequel model by model needed tools.

4.2.2 Step 1: Description of the Problem

Use Case. *In this first step, thematicians aim at providing a description of the problem to be modeled and simulated (the domain model). In particular, the general question driving the creation of this model should emerge from interactions with the* *modeler, but also simplifying hypotheses. Typically the thematician proposes a first description of the system that will be used as a support for the discussion and the collaboration with the modeler. This initial proposition will be modified thanks to modelers' and other thematicians' questions, misunderstandings or technical limitations that must be imposed (for example, in the Microbes project, the modeler cannot let thematicians believe that they could simulate a soil cube with a 1000km edge). This textual description is supported by additional pictures, data and/or articles that could improve the understanding of other stakeholders. An example of this description for the Sworm model is given by the Figure 3. With an informal graphical way, this first model gives an idea of the model functioning. It was the result of exchanges between the biologist and the computer scientist working in the Sworm model. Its establishment facilitated scientific discussions. It permitted to define a design of the intended system which is at the middle of soil-sciences and computer sciences domains.*

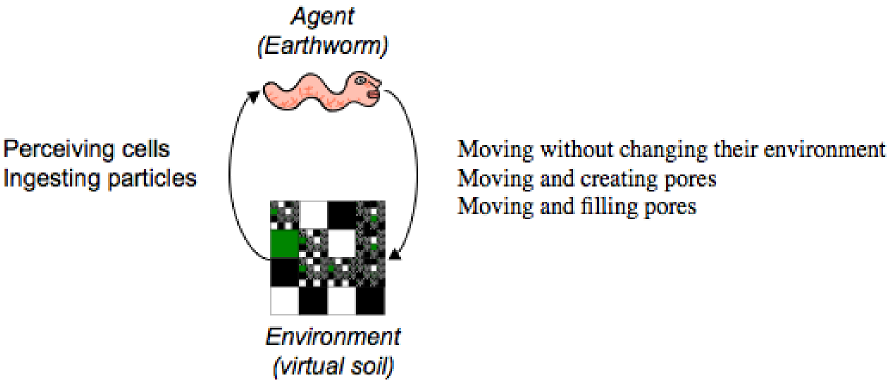


Fig. 3 Ontology description in the Sworm model

By experience an efficient way to process is to use a collaborative editing tool coupled with synchronous communication tools (chat or visio-conference, whiteboard): the thematician's initial proposition is modified in real-time by any participant depending on discussion led in parallel. To improve this step, tools can also help to improve the initial proposition. The editing tool could contain initially a guideline to redact it properly, *i.e.* a framework with various optional questions, remarks or suggestions. We argue that this guideline should be specific to each research domain despite the fact that some questions could be common to must of them.[1] Future investigations and inquiries are thus needed in each community to define these guidelines. The whiteboard has also a big importance: it allows users to draw schema to support a discussion or to summarize in an intuitive way a very big and complex written description of the system. What has been drawn should

[1] For example: what is the biological phenomenon studied? In which kind of project? Which limitations should be imposed to the representation? What is the aim of the modeling? ...

thus be saved and the possibility to insert it into the textual description should be given to users. The portal should thus give the possibility to share various kinds of documents and files and to discuss on them.

To develop such collaborative editing tool we can find inspiration from Google Document[2] or Zoho writer[3] web applications. Note that we think that existing brainstorming tools are not well adapted to this stage. Albeit they are dedicated to take into account collaboration, they are designed to structure reflection from various people sharing the same language, with the same goal and point of view and so on. While trying to describe the problem, thematicians (perhaps from various research domains) and modelers do not have this kind of mutual understanding.

Summary of the needed tools:

- collaborative synchronous and asynchronous **communication tools**
- a collaborative rich-**text editor** with domain specific templates
- a system of **file deposit and sharing**;
- a collaborative **whiteboard**
- a **discussion/commentary system** about the text

4.2.3 Step 2: Description of the Ontology

Use Case. *From the previous description, modelers and thematicians will develop collaboratively a common ontology describing main concepts of the system, their properties, links between them and so on. An example of this ontology description step is to identify concepts used in the Sworm project such as: (i) two kinds of entities, earthworm and soil; (ii) the taxonomy of the interactions between earthworms and the soil; (iii) and what the three behaviours ("Moving without changing their environment", "Moving and creating pores" and "Moving and filling pores") mean.*

For this purpose, we have to integrate collaborative graphical editor to our application. An advanced collaborative whiteboard with specific graphical primitives can be suitable. We could also integrate instead FreeStyler application [14], an application allowing to develop various kinds of schemas, or take a leaf out of it to develop our own application. An interesting way is to have a look to mind-mapping tools, such as XMind[4] or Mindmeister [5]. Actually any tool permitting a great freedom to users in terms of graphical language semantics can be useful to represent the ontological model: no strongly constrained language with heavy semantics is needed. In contrary modelers and thematicians should be able to feel free to draw and link various concepts. We thus argue that as a first attempt a simple tool is enough. It should be able to draw boxes for concepts and arrows (with free labeling) to describe links between these concepts. We could give the capability to users to be more precise about entities by describing their features (states variables). To improve the discussion around the model, stakeholders should be able to draw freely on the schema to

[2] http://docs.google.com/
[3] http://writer.zoho.com/
[4] http://www.xmind.org/us/
[5] http://www.mindmeister.com/

explain their ideas. So the whiteboard should integrate all these features and have a *mind-mapping mode* and a *free mode*.

We could introduce a simple coherence maintenance system between the initial domain model and the ontological model by checking that all concepts appearing in the ontology are also described in the textual description. We think that elements of this ontology should be described as precisely as possible and explained to avoid misunderstandings.

Summary of the needed tools:

- collaborative **communication tools** ;
- a collaborative **graphical tools** with a *mind-mapping* and a *free* mode;
- a **coherence maintenance system** between the domain model and the ontological model;
- a **discussion/commentary system** about the graphics.

4.2.4 Step 3: Definition of the Agents with Their Properties and of the Simulation Parameters

Use Case. *From the above ontology thematicians and modelers can extract entities that will become the agents of the conceptual model. They will also extract properties of these agents from the ontology and the system description. In the Sworm model, modelers, in accordance with thematicians, have decided that earthworms would be represented by two kinds of agent depending on their species (Milsonia Anomala and Eudrilidae) and their size (respectively 2 mm and 0.2 mm diameter sized). They then decided which elements will characterize their earthworms (for example their size, the daily quantity of food ingested, assimilation rate, ...). They have treated similarly the environment. In addition to its participation to the choice of the agents, the thematician is essential to give plausible data for these variables. Unfortunately, data are often lacking and the thematician has to make researches or experiments to determine them.*

At this step, the tool should give to users the opportunity to define agents. The definition of an agent includes its description and the description of its internal state (with a state variables set). To have a better view (and thus a better understanding) of the agents and the relations between them, the experience with Sworm shows that UML class diagrams (or a least a simplified drift) is well suited for this purpose. Nevertheless a parallel representation under the form of a table allows the stakeholders to detail variables (*e.g.* their unit, range, default value...).

Thematicians and modelers also have to define various parameters of the simulation (that are values that can be modified during the simulation process or before the beginning of each simulation, such as the initial number of earthworms in Sworm).

As far as communication is concerned, users should be able to show which parameters they are talking about (by some highlighting process). From an asynchronous point of view, for each parameters, users should be able to post commentary (for example a click on a bouton opens a pop-up to let a message on the forum). They also could be warned if there is unread messages (by a specific icon) on each variable and have the opportunity to read unread messages in a pop-up, without leaving the current page.

Moreover this step will highlight some lacks of data. The tool should thus propose simple feature allowing the modeler to warn a thematician of the lack and keeping track of the warning. For example, for each parameter, we should add a bouton to request additional data to the thematicians. Thus we will integrate a panel summarizing requested data. Note that all these comments and requested data could be recalled to users via a RSS stream.

As far as coherence maintenance is concerned, when a user creates a new entity, the tool will give him the choice between entities defined in the previous model or to use a new name. In the former case, parameters defined in the ontological model will be automatically added to the created agent. Moreover if a parameter is added in this model, it should also be added to the ontological model. In the latter case, the tool should use a graphical symbol to warm the user. Note that this feature will impose additional constraints in terms of control. If a user has the control on the definition of the agents tool, another user controlling the ontological model tool will not be able to delete agents or modify parameters of agents. He will only have the right to create new entities and links and to modify features of entities that have not been defined as agents ; entities having created agents will need special rights or confirmation process to be deleted.

Summary of the needed tools:

- collaborative **communication tools** ;
- a collaborative **graphical tools** for UML class diagram;
- a collaborative interface to detail agents' internal variables and parameters;
- a **coherence maintenance system** between the ontological model and the agent description model;

4.2.5 Step 4: Design of Behaviors

Use Case. *Once stakeholders have defined agents and their variables, they design their behaviors. From information given by the thematician, they extract the main behaviors of each agent (for example in the case of the agent earthworm, the three moving, the perceiving cells and the ingesting particles behaviors) and then begin to refine each behavior. In order to ease the interactions between stackeholders, a graphical language is used to describe behavior. Only after a quite precise specification the computer scientist implements behaviors.*

We argue that by combining advantages of Repast Symphony modeling tool and Modeling4All presented above, we can think to an efficient and role-based behavior design process. Indeed each participant of the project has his own capabilities that should be used as efficiently as possible. To force a domain expert to implement models will induce a big time-loss by learning the programming language or by using heavy and unadapted applications such as pedagogical tools presented above. We argue that it is more efficient to let this task to computer scientists. Our aim is thus to provide a collaborative tool improving the teamwork between expert domain and modeler.

The use of a graphical language to describe behaviors allows expert domains to understand and discuss the general flow of the agents' behaviors. The modeler or computer scientist task is thus to describe graphically the logic of the algorithm and the link between micro-behaviors. We thus only have to describe them without giving the detailed code. Both participants can also make the conception phase together simply by describing micro-behaviors in some natural language that computer scientist will implement himself later. Another benefits of this view is to delay as late as possible the choice of a programming language and thus the model stays as long as possible simulation platform independent. This behavior could be taken from a library of existing micro-behaviors or created from scratch by the computer scientist via a dedicated IDE.

The global description of each micro behavior should thus be made in collaboration between skateholders. It should include a name, a textual description (to define the more precisely possible what the micro behavior should do), the set of variables/parameters that it will use, and which ones it will alter.

Note that the source code does not need to be available to each stakeholder. In particular the thematician does not need to view it. Moreover specific tools for the modeler or the computer scientist could be added to test independently each behavior to debug its source code.

4.2.6 Step 5: Simulation

Use Case. *After a first attempt of implementation of the Sworm model, thematicians, modelers and computer scientists have to test and debug it. The only way is to launch simulations and to observe their results and outputs. Before launching simulations, they have to initialize the global environment (GIS or grid), the agents, the parameters... Then they launch experiments and can explore the simulation by altering parameters values. When an unexpected behavior is observed, thematicians and computer scientists (who well know the source code) can try to find an explanation to this mistake; they will then adjust the model (any model from the description of the problem to the implementation can be modified). Once simulations appear* correct*, thematicians will want to make deeper experiments on the model, for example by testing influence of various parameters on the results.*

As shown previously, it is important that both thematicians and modelers initialize, run and interact with the simulation to be able to discuss results and find together explanation of unexpected phenomenon. They thus need a tool to simulate collaboratively. To this purpose we propose to use the PAMS platform [25] that is presented in details in Section 4.3. Nevertheless PAMS should be extended to support precise initialization and experiment plan definition. As an huge quantity of data is often necessary for initialize real models, the tool should thus allow to import initialization data file. [6] It should also be extended to support the description of an experiment plan, which includes the number of simulation to launch, variation interval for each parameters or the optimization strategy.

[6] An extension of the tool could allow to get data for the world wide web and integrate them in the simulation...

4.2.7 Summary

To summarize the above description, a collaborative Modeling and Simulation platform should at least include following tools (note that all these tools must be collaborative to take into account this particular feature of the Complex System modeling process):

- synchronous communication tools;
- asynchronous communication tools;
- a word processor;
- a diagram creator;
- a behavior designer;
- a simulation manager;
- a simulation control pane.

All these elements are summarized in Table 1. Note that the last four tools should share a common model because each element of the model is linked to other ones. Each tool thus modifies a specific part of the model. We argue also that various models are not fully independent, and some coherence maintenance processes should also be integrated to help users (without being too strict).

The implementation of the portal with all its tools is still ongoing. Nevertheless, an existing portal including communication tools and the simulation control pane has already been developed. Other tools are in development. We describe it in details in the next section.

4.3 PAMS Portal: A Collaborative Simulation Control Pane

In a previous work we had developed a web-portal dedicated to collaborative simulation named PAMS [25]. The key idea subtending this application is to consider the simulator as the collaboration support and thus an object shared by a group of researchers. PAMS allows thus researchers to work collaboratively on a simulator by manipulating, configuring (and in particular by modifying the input parameters), launching it and analyzing the results collaboratively. Figure 5 presents the graphical user interface of the experiment manipulation panel. PAMS supports various generic simulations platforms: GAMA [1], Repast [26], NetLogo [35]. PAMS offers also the advantage to be modular and thus new simulation platforms (such as MadKit [12]) or *ad hoc* simulators (such as Ichthyop [19]) could easily be integrated in. In addition to the specific collaborative tool dedicated to the simulation, PAMS provides essential generic collaborative tools needed to communicate easily (chat, visio-conference, forum...).

PAMS is based on free and open-source technologies. In particular, we used Sakai [1] to manage the collaborative web interface and some asynchronous collaborative tools (such as file management, forum...), whereas Red5[7] deals with synchronous collaborative tools such as chat and visio-conference. From a technological point of view, PAMS uses a standard technology combination: a web application server

[7] http://osflash.org/red5

based on JSP, AJAX and Servlets (on a Tomcat servers), a distributed component based application server (EJB on a Jonas server) to execute simulators and so on and a MySQL database to store results. PAMS has been developed with a modular architecture in order to be easily extended. The architecture has been detailed in [25]. We present the global logical architecture in the Figure 4.

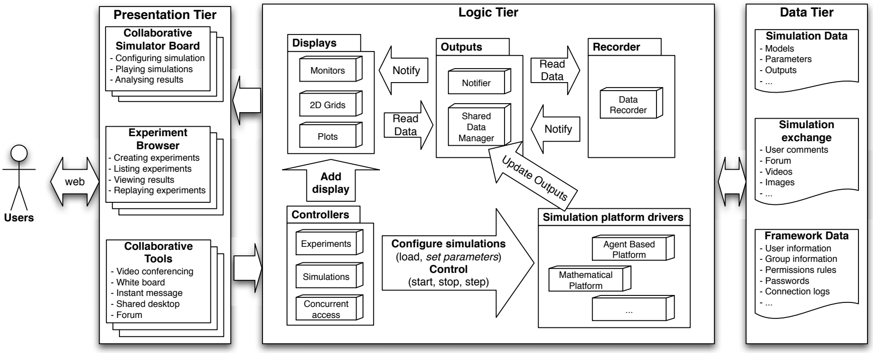


Fig. 4 Global logical architecture of the PAMS portal

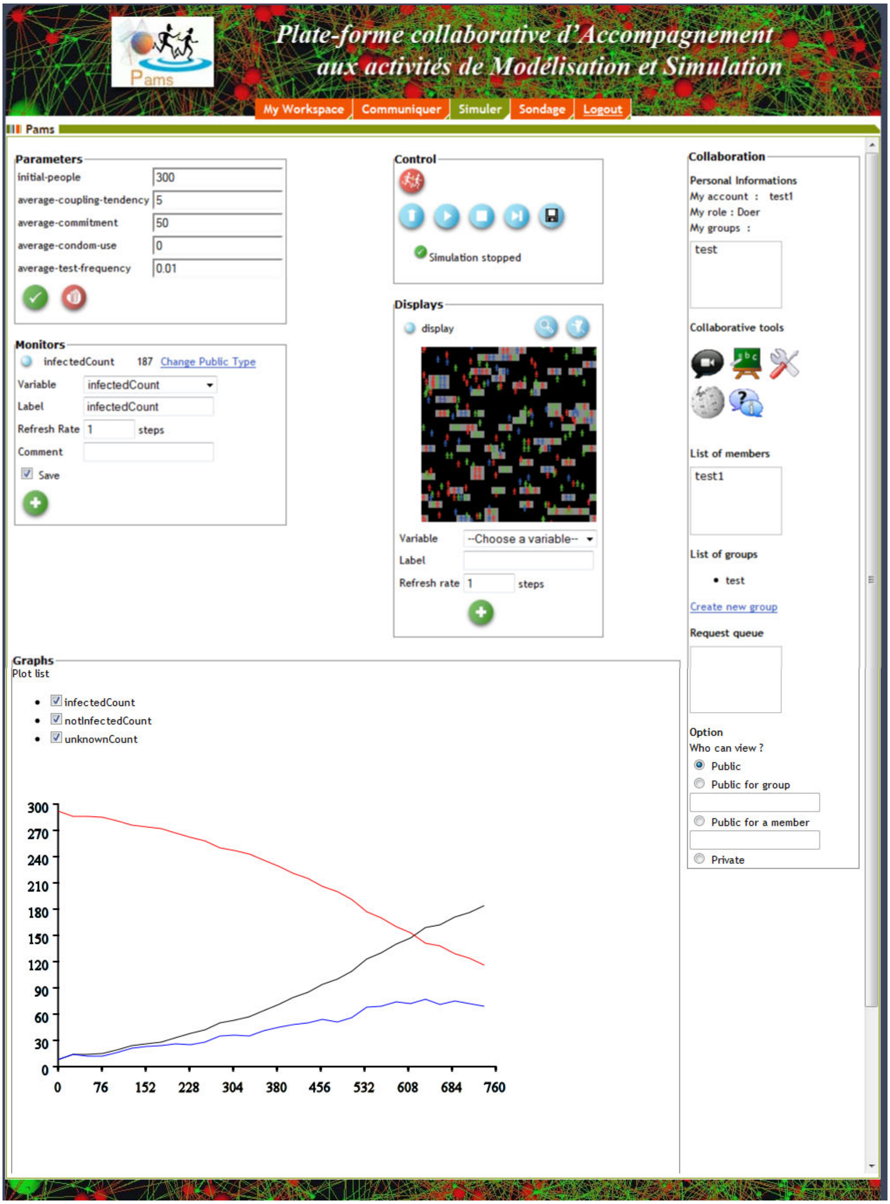
The implementation of the proposed tool will be an extension of the PAMS portal by adding tools implementing functionalities presented above as presented in Figure 4. The main conceptual difference between existing PAMS and our modeling tool is the collaborative support: in PAMS, the simulator is the place of the collaboration and the shared object whereas in the modeling tool, a model is the key object that will be created, manipulated, modified in various way, tested by simulation, annotated...

5 Discussion and Related Work

After having introduced collaboration in the modeling and simulation methodology, and identified features needed to provide it, we now compare our recommendations with other existing collaborative modeling tools and discuss main differences.

5.1 Collaboration in Tools for Modeling and Simulation of Complex Systems

Except Modelling4All [16] presented above, no tool for modeling and simulation of Complex Systems has collaborative features. Moreover with Modelling4All, collaboration is limited only to the sharing with the community of micro-behaviors, which is quite limited. As modeling tools for CS do not have expected collaboration features, we thus need to investigate collaborative tool for modeling in general.


Pams
Plate-forme collaborative d'Accompagnement aux activités de Modélisation et Simulation
My Workspace
Communiquer
Simuler
Sondage
Logout
Pams

Parameters

initial-people	300
average-coupling-tendency	5
average-commitment	50
average-condom-use	0
average-test-frequency	0.01

Monitors

infectedCount 187 Change Public Type

Variable	infectedCount
Label	infectedCount
Refresh Rate	1 steps
Comment	

☑ Save

Control

Simulation stopped

Displays

display

Variable	--Choose a variable--
Label	
Refresh rate	1 steps

Collaboration

Personal Informations
My account : test1
My role : Doer
My groups :
test

Collaborative tools

List of members
test1

List of groups
- test

Create new group

Request queue

Option
Who can view ?
● Public
○ Public for group
○ Public for a member
○ Private

Graphs
Plot list
- ☑ infectedCount
- ☑ notInfectedCount
- ☑ unknownCount

Fig. 5 PAMS portal GUI

5.2 *Collaboration in Modeling Tools and Methodologies*

5.2.1 Conversational Modeling

[33] proposes to split the modeling process in four distinct methodologies (perception of the system and static models; design of simulation models; implementation, debugging and validation; inquiry into the simulation) and detailed requirements for the first one. Authors propose to use the Conversational Modeling application [32] to handle the first step. This tool allows stakeholders to link and manage their ideas and thus improve discussion between them.

The case study presented in [33] is very interesting because it gathers experts with various competence domains to take part to the modeling process. Stakeholders have thus to actually collaborate to elaborate the model. Nevertheless in the use case, all experts are in the same room and the application is only here to support and summarize the discussion. Our methodology follow the same idea of a collaborative modeling with various expert domains, but our approach is dedicated to geographically separated researchers. It should thus propose additional communication tools and awareness mechanisms. The model is not a help improving interactions between stakeholders but the center of their work.

5.2.2 COMA: A Collaborative Tool Dedicated to the Negotiation in the Modeling Process

The COllaborative Modeling Architecture tool (COMA tool) [30] is a collaborative tool dedicated to the information synthesis and the negotiation in the process of collaborative modeling. It allows modelers to propose their own model, to support or challenge proposed model and provide various rules and processes to decide the accepted proposal.

It is important to note that this tool does not allow modelers to synchronously discuss on the model and to modify it: their only choices are the defense or the challenge of the model as a whole. Our experience in the domain of modeling and simulation of Complex Systems shows (as in the Sworm model project) that thematicians and modelers often work together on the model. They improve it incrementally by taking advantage of the multiplicity of viewpoints and skills. Moreover continuous interactions are sometimes necessary in cases where there is not one stakeholder that have a complete view of the system. That are the reasons why we favor the direct collaboration on models rather than the offline and asynchronous model creation in a first step and its negotiation in a second step.

5.2.3 Tools Extending Software Configuration Management System

Some researches take as startpoint the fact that software development is more and more located in emergent countries where the working force is really cheap. But this outsourcing induces the need of new tools to allow remote and collaborative software development between geographically separated programmers. Conceptually such tools extend Configuration Management system such as CVS and Subversion.

[3] presents the CoDesign framework. This framework proposes a conflict detection framework. In particular, it allows to detect modeling conflicts such as synchronization conflicts but also both syntactic and semantic conflicts. [9] proposes the DISCOM framework mainly based on the idea that it is more efficient to store only modifications from one version to another one than to store both whole models.

Whereas CoDesign is focused on the conflict detection between distinct code developed by various and geographically separated programmers, our approach is focused on the simultaneous modification of a model. Nevertheless, as our whole model is composed of various submodels that can be developed in parallel, some incoherence could appear. This is the reason why we have introduced a system of coherence maintenance between models. It could also be interesting in a future extension of our tool to add the possibility for modelers to compare various versions of the same model to appreciate improvements and modifications made between two versions.

Collaborative modeling tools that we are aware of are mainly tools dedicated to software modeling. The main drawback of these approaches and tools is that they are only dedicated to very particular tasks of the modeling and simulation process. Our methodology aims at covering the whole process and thus has to tackle much more issues.

6 Conclusion

The Modeling and Simulation process of Complex Systems is by nature a collaborative task because it involves people with complementary skills and knowledge. Lot of methodologies have been proposed to drive the modeling process. But only few have addressed the fundamental issue of the collaboration and taken it into account. In this paper we have extended Drogoul *et al.*'s methodology by describing interactions and collaborations between modeling participants. We then have proposed and described the minimal set of collaborative tools needed to support this method.

We have only proposed here a minimal set of tools. Of course we need to make deeper investigations and observations of actual modeling and simulation projects to provide the complete list of tools needed to support such a project. Moreover once this conception phase will be over, we have to implement the platform. The implementation will be integrated into the existing collaborative simulation portal PAMS [25]. Our platform should then be extensively tested with thematicians. We also aim to use our platform to evaluate benefits of such a platform for modeling projects.

Acknowledgements. This work was funded by the French IRD SPIRALES research program.

References

1. Allan, R., Yang, X., Fish, A., Gonzalez, M., Crouchley, R.: Using sakai for e-research: Building a multi-institution virtual research environment (2007) (presentation)
2. Amouroux, E., Quang, C., Boucher, A., Drogoul, A.: GAMA: an environment for implementing and running spatially explicit multi-agent simulations. In: 10th Pacific Rim International Workshop on Multi-Agents (PRIMA), Thailand (2007)
3. Bang, J.y., Popescu, D., Edwards, G., Medvidovic, N., Kulkarni, N., Rama, G.M., Padmanabhuni, S.: Codesign: a highly extensible collaborative software modeling framework. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, Cape Town, South Africa, vol. 2, pp. 243–246 (2010)
4. Bauer, B., Müller, J.P., Odell, J.: Agent UML: A formalism for specifying multiagent interaction. In: AOSE, pp. 91–103. Springer, Berlin (2001)
5. Beck, K.: Extreme Programming Explained. Embrace Change. Addison-Wesley Professional, Reading (1999)
6. Bird, N., Perrier, E.: The pore-solid-fractal model of soil density scaling. EJSS 54, 467–476 (2003)
7. Blanchart, E., Marilleau, N., Chotte, J., Drogoul, A., Perrier, E., Cambier, C.: SWORM: an agent-based model to simulate the effect of earthworms on soil structure. European Journal of Soil Science 60(1), 13–21 (2009)
8. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: TROPOS: An agent-oriented software development methodology. Journal of Autonomous Agents and Multi-Agent Systems 8(3), 203–236 (2004)
9. Cicchetti, A., Muccini, H., Pelliccione, P., Pierantonio, A.: Towards a framework for distributed and collaborative modeling. In: WETICE, pp. 149–154 (2009)
10. Drogoul, A., Diane, V., Meurisse, T.: Multi-agent based simulation: Where are the agents? In: Sichman, J.S., Bousquet, F., Davidsson, P. (eds.) MABS 2002. LNCS (LNAI), vol. 2581, pp. 1–15. Springer, Heidelberg (2003)
11. ElliS, C., Gibbs, S., Rein, G.: Groupware some issues and experineces. Communications of the ACM 34(1), 38–58 (1991)
12. 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)
13. Fishwick, P.: Simulation model design and execution. Prentice Hall, Englewood Cliffs (1995)
14. Giemza, A., Ziebarth, S.: Overview of the freestyler modeling environment. Tech. rep., Collide, University of Duisburg-Essen (2008)
15. Gilbert, N., Troitzsch, K.G.: Simulation for the Social Scientist. Open University Press, Stony Stratford (1999)
16. Kahn, K., Noble, H.: The Modelling4All Project - A web-based modelling tool embedded in Web 2.0. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, SIMUTools 2009 (2009)
17. Kay, A.: Squeak etoys, children & learning. Tech. Rep. RN-2005-001, ViewPoints Research Institute (2005)
18. Klopfer, E., Begel, A.: StarLogo TNG. An Introduction to Game Development. Journal of E-Learning (2005)
19. Lett, C., Verley, P., Mullon, C., Parada, C., Brochier, T., Penven, P., Blanke, B.: A lagrangian tool for modelling ichthyoplankton dynamics. Environmental Modelling & Software 23(9), 1210–1214 (2008)

20. Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., Resnick, M.: Scratch: A sneak preview. In: Second International Conference on Creating, Connecting, and Collaborating through Computing, Kyoto, Japan, pp. 104–109 (2004)
21. Marilleau, N.: Méthodologie, formalismes et outils de modélisation-simulation pour l'étude des systèmes complexes: application à la mobilité géographique. Ph.D. thesis, l'Université de Franche-Comté (2006)
22. Marilleau, N., Cambier, C., Drogoul, A., Perrier, E., Chotte, J., Blanchart, E.: Multiscale MAS modelling to simulate the soil environment: Application to soil ecology. Simulation Modelling Practice and Theory 16(7), 736–745 (2008)
23. Muller, J.P.: The MIMOSA generic modeling and simulatiion platform: the case of multiagent systems. In: Coelho, H., Espinasse, B. (eds.) 5th Workshop on Agent-Based Simulation, Lisbon, Portugal, pp. 77–86 (2004)
24. Muller, J.P., Ratzé, C., Gillet, F., Stoffel, K.: Modeling and simulating hierarchies using an agent-based approach. In: Zerger, A., Argent, R.M. (eds.) MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, Australia (2005)
25. Nguyen, T.K., Gaudou, B., Ho, T.V., Marilleau, N.: Application of PAMS collaboration platform to simulation-based researches in soil science: The case of the micro-organism project. In: IEEE-RIVF International Conference on Computing and Telecommunication Technologies (IEEE-RIVF 2009), Da Nang, Vietnam (2009) (to appear)
26. North, M., Collier, N., Vos, J.: Experiences creating three implementations of the repast agent modeling toolkit. ACM Transactions on Modeling and Computer Simulation 16(1), 1–25 (2006)
27. North, M., Howe, T., Collier, N., Vos, J.: The Repast Simphony Development Environment. In: Macal, C., North, M., Sallach, D. (eds.) Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms (2005)
28. Occello, M., Koning, J.L., Baeijis, C.: Conception des systèmes multi-agent: quelques éléments de réflexion méthodologique. Technique et science informatique 20(2), 233–263 (2001)
29. Repenning, A.: Agentsheets: A tool for building domain-oriented dynamic, visual environments. Ph.D. thesis, University of Colorado (1993)
30. Rittgen, P.: Coma: A tool for collaborative modeling. In: Bellahsene, Z., Woo, C., Hunt, E., Franch, X., Coletta, R. (eds.) Proceedings of the Forum at the CAiSE 2008 Conference, Montpellier, France. CEUR Workshop Proceedings, vol. 344, pp. 61–64 (2008)
31. Seals, C., Rosson, M.B., Carroll, J.M., Lewis, T., Colson, L.: Fun learning Stagecast Creator: An exercise in minimalism and collaboration. In: Proc. IEEE Symp. on Human-Centric Computing 2002, pp. 177–186. IEEE, Arlington (2002)
32. Selvin, A.M.: Conversational modeling: A software-supported technique for analysis by teams. Tech. Rep. TM-960045, NYNEX Science & Technology Technical Memorandum (1996)
33. Sierhuis, M., Selvin, A.M.: Towards a framework for collaborative modeling and simulation. In: Proceedings of Workshop on Strategies for Collaborative Modeling & Simulation (1996)
34. Susi, A., Perini, A., Mylopoulos, J.: The TROPOS metamodel and its use. Informatica 29, 401–408 (2005)
35. Wilensky, U., Evanston, I.: Netlogo. Center for connected learning and computer based modeling. Tech. rep., Northwestern University (1999)
36. Wooldridge, M., Jennings, N.R., Kinny, D.: The GAIA methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems 3, 285–312 (2000)

Role-Based Collaboration Extended to Pervasive Computing

Kahina Hamadache and Luigi Lancieri

Abstract. Computer Supported Collaborative Work (CSCW) has been a hot point in researches for some decades. Recent progresses in software and hardware technologies have allowed the use of more and more advanced applications and services. It also brought computing capabilities to mobile devices such as smartphones and laptops. This has led to an extensive use of computers to collaborate in some unexpected manners. Among the abundance of models designed to support collaboration some are particularly promising: tasks models, roles models and collaboration's context models. Simultaneously the Pervasive Computing paradigm has emerged from recent researches. In this paper we propose a model to integrate the pervasive computing perspective into the collaborative work. This integration is proposed by the use of an original model: the PCSCW model (Pervasive Computing Supported Collaborative Work). This model relies on some robust concepts: a role model inspired by some recent works, a classical task model coupled to a precise resource model and the development of device collaboration rules. The resulting model provides a seamless and transparent cooperation of devices to simplify and facilitate the collaboration of humans.

Keywords: Pervasive Computing, Collaborative Work Modelling, Role-Based Collaboration Model.

1 Introduction

In the past few years, computer supported collaborative work (CSCW) has became an unavoidable aspect of everyday's life of most companies. Actually, computers

Kahina Hamadache
Orange Labs, 42 rue des Coutures, 14000 Caen, France
e-mail: kahina.hamadache@orange-ftgroup.com

Luigi Lancieri
LIFL, University Lille 1, Lille, France
e-mail: luigi.lancieri@univ-lille1.fr

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 55–78.
springerlink.com

can channel the collaboration between people in many ways; they facilitate many tasks and allow monitoring and regulating of the collaboration. Most simple tools for collaboration already exists, even if those are still young and lack of maturity they offer the basis for simple collaboration. However, to provide more advanced collaboration features and opportunities, it is necessary to go deeper in this research field.

Research in CSCW is an “old” domain (relatively to computer sciences history) and many researchers have focused on the design of collaboration model with the final objective to allow a better management and understanding of collaboration. Thus many interesting models have been carried out in the history of collaborative software, but as for now none of them have ever really be able to overcome others and to solve all issues. However we think that some models are more relevant and promising. This is the case for role-based models. These models focus on the simple but still unavoidable concept of roles in collaboration. A role can be defined in terms of responsibilities and rights where responsibilities are actions a role player must perform and rights are actions he can perform. Zhu and Seguin [13] characterise a role in collaboration as follows:

- A role is independent of persons;
- A role should consider both responsibilities when the human player is taken as a server and rights when the human player is taken as a client.
- A role has to accomplish the tasks specified by the responsibilities.
- A role can be performed by one or many human players at the same time. A role can be created, changed and deleted by a human user with a special role.

Another evolution we can observe nowadays is that electronic devices are becoming more and more common, compact, smart and autonomous. Thus, the digital environment is taking each day a greater part in our world, leading us to a digital augmented environment. In this vision, the pervasive computing paradigm is a major domain, it aims at making all these smart electronic devices spread in our environment collaborate to provide us a seamless interaction with the digital world.

A natural extension for CSCW is then to be more suited for this kind of environment where the number and the type of devices is nothing more than a variable to which you have to be able to adapt. In this perspective, an interesting challenge for CSCW is to know dynamically and efficiently how to take advantages of this technology to improve the collaboration between humans. Potential advantages are various, we may think to simple mechanisms such as smart device control via a computer until complex mechanisms of collaborative context awareness.

However, only few advances have been made toward the better integration of pervasive features in CSCW systems. Thus the main objective of our work is to provide a model to natively support the pervasive computing paradigm inside the collaboration of users. As we will see in the next sections this model relies on several sub models: a role model, a task model and a resource model. Those models are coupled with the definition of “device collaboration rules” that define automatic device behaviours according to a given context.

In the remainder of this paper, we present our role-based model for pervasive computing supported collaborative work. We start with a presentation of already existing collaboration models for CSCW based on tasks, roles and collaborative awareness. Once all this notions have been introduced we are able to present our Pervasive Computing Supported Collaborative Work model, its main principles and two illustrative use cases. Following our researches chronology we try to anchor or model in a more concrete perspective by presenting its dedicated simulator, this last point allows us to introduce basics about CSCW evaluation and the way we consider it to validate and evaluate our model.

2 Collaborations Models for CSCW

The main purpose of CSCW systems is to handle the collaboration between users. To do it they can rely on technical evolutions and tools adaptation for multiple users. But collaboration raises problems that are going far beyond technical issues. Indeed, the main problem of CSCW is the collaboration itself, how a system can effectively support collaboration patterns and how it can be aware of the current collaboration status. To tackle these problems several models of collaboration have been proposed. Among them, some kinds sounded more promising: tasks models, roles models and the refined collaborative awareness models. In the following we will illustrate these models by presenting some of their related researches.

2.1 Task Models

The task model is now widely accepted by the CSCW community to be one of the bases to represent collaborative work. It has been the subject of many articles and is still an active research field. Task models' goal is to identify useful abstractions highlighting the main aspects that should be considered when designing interactive systems. The main advantage of task models is that they represent the logical activities that an application must support in order to better understand the user, his work and his expectations.

Keeping in mind the necessity to improve the general ergonomics of collaboration, Molina et al [4] proposed a generic methodology for the development of groupware user interfaces. This approach is called CIAM (Collaborative Interactive Applications Methodology); the approach defines several interrelated models capturing roles, tasks, interactions, domain objects and presentation elements. Even though, the models cover all important aspects involved in groupware user interfaces, they are only used at the analysis stage. Subsequent development phases (e.g. requirements or design) are not covered. The methodology is not assisted by a tool which would facilitate the creation and simulation of the various models. In particular, the latter is an important shortcoming since the animation of models is an important technique to obtain stakeholder’s feedback. These works fail to account for user roles and multiple role-based views on the same collaborative task. Aiming at fulfilling this gap Vellis [11] has adopted an extended version of CCTT (Collaborative ConcurTaskTrees [16]), which would be taking care of user role differentiations and their effect in the whole process.

In a different perspective Penichet et al [9] propose a task model for CSCW based on the use of several well known task modeling aspects. Their model is aimed at describing "the tasks that should be performed to achieve the application goals" by giving them a good characterization. This model is aimed at designers that have to design groupware systems. What they propose is not a complete new model of tasks but a new "composition" of existing tasks models in order to have a better, more complete and more effective task model. Their approach is based on the description of tasks that are realized in groupware systems keeping in mind more classical aspects and mechanisms to analyze them. They argue that classical CSCW features or time-space features are not enough to correctly describe a groupware, but that a well done combination of them can do it.

Task models are interesting, because they can be easily understood by humans as they represent "classical" organization of collaborative work. But some models take different ways to represent the collaboration, making them interesting by the simple fact that they have new points of view of the collaboration.

2.2 Collaborative Awareness Models

Collaborative awareness is the capacity to be aware of the current state of the collaboration. It can be useful for humans when they need to coordinate themselves to accomplish a specific task or schedule an operation. But we know it can also be useful for devices in order to behave properly. Thus collaboration awareness is critical for most of collaborative systems as it helps maintaining the coherence of the collaboration and eases the work of collaborators. Let's have a look at some researches of this domain.

Drudy and Williams [17] proposed a cooperative awareness model based on role, but the relation in roles' cooperation was not mentioned in their paper. Gutwin and Greenberg [18], proposed a workspace awareness framework, this framework describes three aspects: its component elements, the mechanisms used to maintain it, and its uses in collaboration. These parts correspond to three tasks that the groupware designer must undertake in supporting workspace awareness; understand what information to provide, determine how the knowledge will be gathered, and determine when and where the knowledge will be used.

In an analytic approach Yan and Zeng [10], [20] proposed an original model for group awareness inside CSCW systems. They assume that there are mainly two aspects in group awareness: "group awareness model" and the "method of realization". They point out the fact that for now, the main problem is still the construction of a robust model.

What they want to do to solve the resisting problems is to analyze basic elements of group work: "task", "action" and "role".

The application they developed is composed of a set of modules, each one dedicated to a specific goal, but the more interesting of them is obviously the task disassembling one. The elementary definition here is the formal definition of a task as a triplet *T: (Role1, Action, Role2)* where T is the task, Role1 and Role2 are roles associated with the task. Role2 is only mandatory when Role1 cannot complete independently the Action. So, to disassemble a task, the system recursively disassemble Role1, Action and Role2 until it can't divide any more. When it reaches this state the task is defined as "atomic". They also define a set of rules for disassembling in order to avoid inconsistent state. Moreover they notice that task disassembling is time consuming and then propose to pre-process most common tasks categories into task tree templates.

Thus they provide templates to represent tasks and then user's activity, allowing them to have an interesting description of current collaboration.

Moreover they reasoned by telling that as task depends on role, task is a more efficient group awareness descriptor. But they also showed that task depends on action. If we follow their argumentation we should say that action is an even more effective descriptor, than they do not mention. Finally, we do not really agree to the task representation they propose. By example we can say that, depending on the group in charge, a task will not be handled the same way, with the same roles and actions, even if the goal is the same. That's why we think that task and roles should not be depicted by some stilling trees and templates but that it should be dynamically extracted and modified during the group collaboration.

While this precedent work relies on the precise dissection of tasks and roles, Rodden [12] proposes a model of awareness for cooperative applications measuring the awareness intensity by the flow of information between application programs. If this model successes in describing group awareness characteristics, it does not really include group structure into its measure.

Researches often want to be as generic as possible, in order to produce a model able of representing any kind of collaboration. Thought, generic implies less coupling with the domain, then most of the times it is necessary for models to focus on a specific domain.

We are reaching the roles models, which are the ones that motivated use for this research. Roles can seem simple, but describing them correctly with all their characteristics is a really complex issue.

2.3 Role-Based Collaboration

Role-based collaboration (RBC) is a methodology to design and implement new computer-based tools. It is an approach that can be used to integrate the theory of roles into CSCW systems and other computer-based systems. It consists in a set of concepts, principles, mechanisms and methods [1]. RBC is intended to provide some benefits to long-term collaboration: identifying the human user "self", avoiding interruption and conflicts, enforcing independency by hiding people under roles, encouraging people to contribute more and removing ambiguities to overcome expectation conflicts. It is also intended to provide benefits to short-term collaboration: working with personalized user interfaces, concentrating on a job and decreasing possibilities of conflicts of shared resources, improving people's satisfaction with more peoples' playing the same role during a period and transferring roles with requirement of a group. Finally, in management and administration, it helps at decreasing the knowledge space of searching, creating dynamics for components and regulating ways of collaboration among parents.

Some CSCW systems have indeed applied the concept of roles. Barbuceanu et al [6] have proposed role based approaches to agent coordination. This approach includes a "practical, implemented coordination language for multi-agent system development" that defines, agents, their organization and roles. Agents play roles in an organization, and a role is defined by its major function, permissions, obligations, and interdictions. A role's permissions include agents under its authority and its acquaintances. An agent's beliefs and reasoning are partitioned on the basis of the roles it plays to facilitate context switching [6]. A combination of events leads to a situation for the organization, with each agent member in a given local state. An agent's behavior in a situation is determined by its conversation plans, and these are usually specified to be between a particular pair of roles.

Edwards [7] propose a system that can implement a variety of useful policies in collaborative settings, particularly in the areas of awareness and coordination. This system uses the notion of roles to associate categories of users with particular policies. Intermezzo roles can represent not only groups of users, but also descriptions of users in the form of predicates evaluated at runtime to determine group membership. Dynamic roles, in particular, expand on one of the central themes in this work: by bringing information about users and their environments into the system, it can make computer augmented collaboration more responsive, and can free users of many of the implicit burdens in working with today's collaborative systems.

In a more recent article, Zhu [2] proposes his view of collaborative authoring based on the use of roles. He points out the fact that collaborative systems should not only support virtual face-to-face collaboration between distant people, but should also improve physical face-to-face by providing mechanisms to overcome drawbacks of face-to-face collaboration. They notice that WYSINWIS (What You See Is Not What I See) can be an efficient model for the development of collaborative systems. Thus WYSINWIS systems can allow different users to have different views of a shared document according to their roles inside the collaboration. This kind of interaction is not totally new, and such systems exist for a long time, but what they propose is a mechanism based on the precise role definition and specification to allow roles to be dynamically tuned and managed in the system.

Furthermore, Zhu and Tang [3] propose a role based hierarchical group awareness model (RHGAM). Firstly RHGAM constructs a group cooperation environment (GCE), and then GCE is extended by group awareness content, awareness hierarchy, the task decomposition rule. The model divides the awareness information into four levels by decomposition and recombination using a role-task graph and the thinking of group structure. In RHGAM, role is the basic of group cooperation; with the different group structure and task relation, the awareness information is shared between roles hierarchically.

In a relatively different perspective, Ahn et al [5] implemented a role-based delegation framework to manage information sharing (FRDIS) for collaborating organizations. Their central idea is to use delegations as a means to propagate access to protected resources by trusted users.

Role models propose a "natural" approach to collaboration; with the help of task models it is possible to have an accurate description of user's collaborative work. Still we want to go further and properly consider and integrate devices as part of the collaboration. To do it we propose our model based on the description of roles, tasks, actions, resources to perform them and available devices' resources.

3 PCSCW Collaboration Model

Keeping in mind the works that have been done in the different domains we're interested in, we propose our own model relying on some simple concepts: tasks, actions, roles and resources. The main principle of this model is the following: we rely on the fine description of roles, tasks, actions, resources required and the available devices' resources; then by a simple comparison of required and available resources we can select the right "device collaboration rule" to make devices collaborate seamlessly and facilitate the collaboration of users. We'll now give further details about these main aspects and their use in the process of making devices automatically and smartly collaborate.

3.1 Task

The first concept to define is the task. This concept is one of the most popular of the recent researches in collaboration modelling. A task can be defined as a set of actions to be performed in a specified or unspecified order to fulfil the task objective. In addition, a task is not always (and in fact most of the time is not) an atomic one, meaning that it can be composed of several sub-tasks with their own actions and objectives. Moreover, we can point out that the collaboration of people takes place when they need to perform a task they can't or shall not do alone. If this task has to be performed by more than one person, it can be considered has a "shared task" or a "common task".

3.2 Actions

Actions can be seen as tasks components. In some perspective they could be considered as atomic tasks, however we think that a task carries its own meaning, actions don't, and that's why we should consider them as sub-atomic tasks. To illustrate this idea we can figure that the action "opening a web browser" has no "meaning", but opening a web browser and writing a word in a search engine has its own meaning, it is the task of "searching on the web".

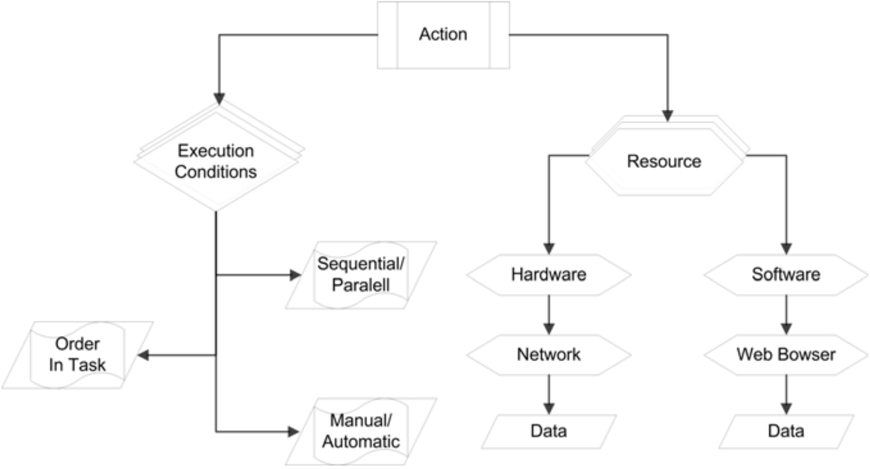


Fig. 1 Action Specification

3.3 Role

As we have seen previously, a role can be defined as a set of tasks to be performed by a single entity, giving it responsibilities, rights and duties. A role is not reserved to persons; it can also be played by a group of persons or by an entire organization. Besides, in the same way as a role can be designed for more than one person, a person can play several roles at a time. This is particularly true in the case of a person belonging to multiple groups (for example a work team and a sport club). In addition, a role can have a specific "cardinality" inside of a group, meaning that you can have several people in the same group playing the "same" role. This aspect of the role concept can be confusing if you consider that two people never do the exact same work, that's why roles have not to be confused with peoples.

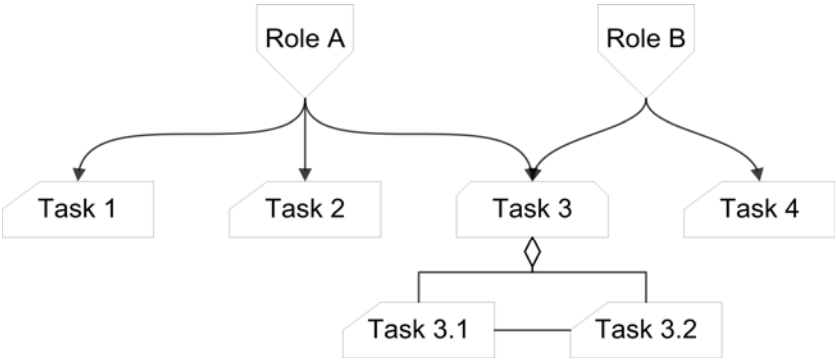


Fig. 2 Roles and Tasks

On figure 2 we give an example of how roles can interact through their allocated tasks. “Role A” has 3 tasks: 2 are dedicated to it and the third is shared with “Role B”. Then this task two is subdivided in 2 subtasks: “Task 3.1” for Role A and “Task 3.1” for Role B. One could have proposed to remove Task 3 and just leave tasks 3.1 and 3.2 affected to their roles. Still we argue that for some of them it is necessary to preserve links between related tasks. Indeed some tasks require several roles to be completed. For instance the task “writing software specification” is composed of two subtasks: “writing software business specifications” and “writing software technical specifications”. Besides, you can’t write technical specifications before business ones have been written and they can’t be always written by the same person, then the two roles associated to this task will not be held by the same person.

3.4 Resources

If you intend to model the context of people in order to develop context awareness mechanisms, at some point of your reflection you will have to face the representation of users’ resources relevant for the part of context you’re interested in. Obviously in our model, we can’t avoid this part, it is in fact one of the most interesting point we want to explore. Indeed, we argue that the description of tasks should be made through the representation of resources required to perform it. Going even further we could describe facultative resources that can be effective to perform the task but which are not mandatory. Thus, considering that you’ve got a fine description of the task a group is performing, you can have accurate indicators of the state of the task. This could lead to a fine monitoring of the task and then to a fine collaboration awareness mechanism.

3.5 Smart Devices

By extension of the precedent aspect of our model, we propose to associate tasks to (smart) devices. To do it we have to figure out that smart devices are parts of the available resources. Furthermore, it is necessary to have a description of devices capabilities. For instance, if you consider that a high-speed connection to the Internet is required for your task, the best device to support it can be quickly identified by a simple query to available ones or by a more efficient request to some kind of a context manager. Such a mechanism is particularly effective, as it can make several devices cooperate seamlessly.

3.6 Roles for Devices

As a refinement of devices description, we propose to define their roles in the collaboration process and more precisely for a task. Thus, a task becomes the natural link between peoples and devices via the description of roles for both of them. As for “Human Role”, the role of a device describes its responsibilities and rights. To take a simple example a device can have a role in a collaboration process giving the responsibility of providing the Internet connection for a given user. With this example we can point out a major difference between human roles and devices roles, humans' roles are based on actions performing while devices roles are based on resources providing. The Fig 3 sums it up. On this figure we quickly modelled a simple, but common task: the development of a client-server application which implies the development of a shared object: the communication interface between

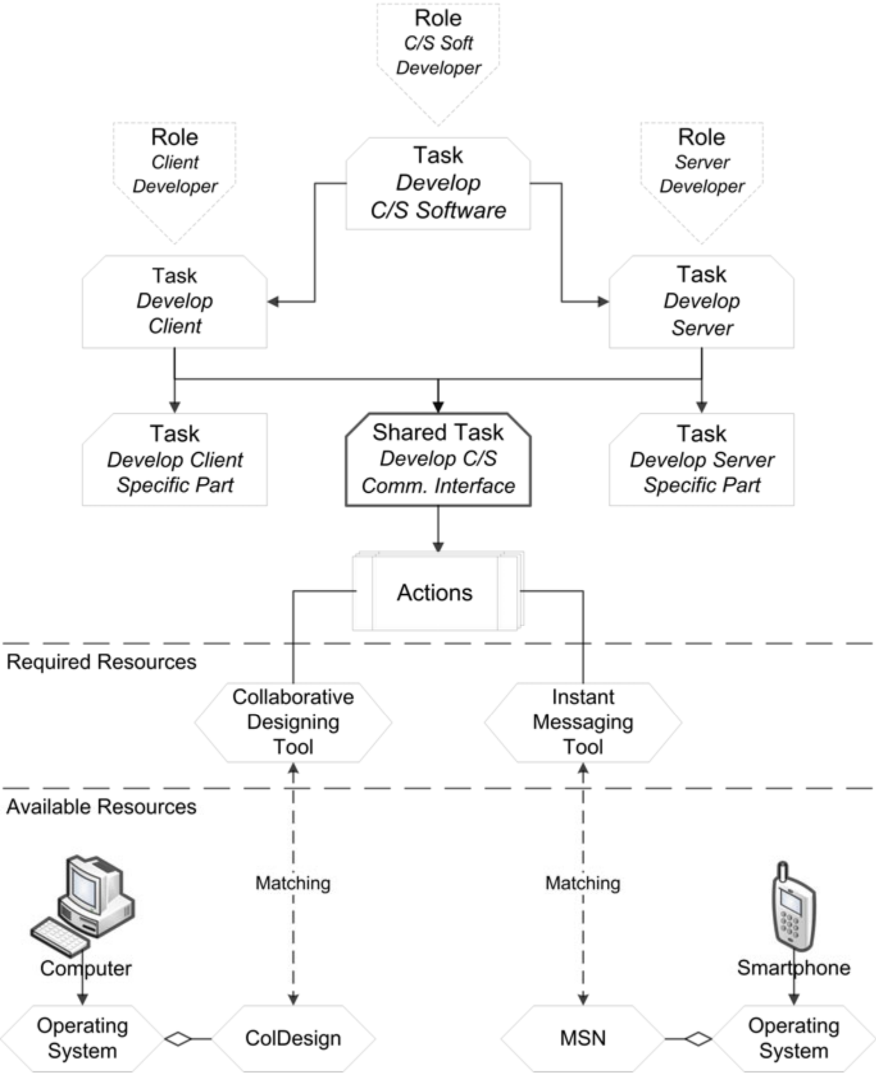


Fig. 3 PCSCW model, example of application

the client and the server. For this task we need a Collaborative Design Tool and an Instant Messaging Tool, which is not mandatory but can improve the collaboration. An interesting point here is that the model can enhance the collaboration by proposing optional resources such as, in this case, a messaging tool. Furthermore the model itself can be refined by describing precise rules for the messaging tool to be proposed and used; in some cases it can be preferable not to use it. The designing tool is provided by a computer while the messaging one is available on the smartphone. Thus we can say that the computer plays the role of "Heavyweight application provider" while the smartphone has a "Messaging application provider" role.

3.7 Devices Collaboration Rules

All devices don't natively support collaboration with others. In order to solve this kind of issue we argue that the definition of device collaboration rules could be of great help. These rules intend to define tasks that could be automatically performed by devices to collaborate in order to allow a user to do its own task. The main idea behind this is the following: a user needs two (or more) resources to complete an action related to a task; these resources are not available on a single device, but the combination of several of them can supply the resources. Thus, device collaboration rules define what actions can be performed by devices to collaborate, finally providing required resources to the user. These collaborations can be of various kinds: network access sharing, heavy computing task delegation and notification of events, anything you can imagine to make several devices cooperate. In a previous paper [19] we defined that each device collaboration rule is written with the following syntax (1):

$$\text{IF } (\mathit{context.resources} \equiv \mathit{rule.resources}) \text{ THEN DO } \mathit{rule.behavior} \qquad (1)$$

This simple syntax summarizes how device collaboration rules are working. The first part of this formula corresponds to the comparison between the current state of the context and resources required to trigger the device collaboration. The second part of the formula corresponds to the collaborative behaviour of devices. Thus if context resources are matching rule resources we trigger the related collaboration of devices. Even if this basic mechanism is a critical part of our model we've established in [19] that it is not sufficient to completely manage. Indeed if we consider the fact that several rules can have the same set of required resources to be triggered or that a single rule can trigger several behaviours at once, we need another mechanism to decide which option has to be used. In order to solve this issue, we propose to define constraints on resources as triplet {P, V, C} where: *P* a parameter which represents the precise point to be evaluated, *V* the expected (or required) value (or threshold) for this parameter and *C* the criticality of this parameter which represents the relative importance of this constraint. In order to facilitate the understanding and the use of criticality we have defined standard levels of criticality: *Optional*, *Very Low*, *Low*, *Average*, *High*, *Very High* and *Mandatory*. It allows us to quantify, estimate, compare and then choose between several candidate rules. In addition to this triplet we propose to organize constraints in five main categories, facilitating and guiding rules designer in their work: *Availability, Cost, Privacy, Reliability and Security*. These constraints describe desired characteristics of resources according to the tasks to be performed. Given these constraints we've got a mean to know which behavior is the most suited in the current state of the context. Thus to decide the most adapted we only have to evaluate the suitability of the potential rules according to their respect of the defined constraints. We will illustrate this mechanism in the second example of the following section.

4 PCSCW in Action, Two Use Cases

To illustrate our model, we can consider a use case that we already mentioned in [8]. This example relies on the Pervasive Brainstorming systems we developed in this previous article and can be described as following:

- The manager of a team wants to have the opinion of its team members about a specific topic (for example about a project he's planning);
- As his team is often spread over different locations, he can't meet each of them physically;
- To solve issues they can encounter for this collaboration, we proposed a service based on the use of mails and forum to channel the opinion of the team and provide efficient synthesis of the group opinion;
- The system itself is based on the automatic publication of multiple-choice questionnaires which can be sent by mail to a dedicated mailbox, mails are then analyzed and contained questionnaires are published on a forum where team members can vote and give their opinion.

If we consider this use case with our model we can distinguish two roles: the manager role and the basic team member role. The manager role has a cardinality of 1 while the team member role has an unspecified cardinality for this group. The team itself is mapped to a "Group Role" with its own set of tasks. The manager role allows its player to perform a "Send New Questionnaire" task while team members are allowed to perform the task "Answer a Questionnaire".

Let's consider the case where a member of the team, Bob, is equipped with a laptop and a cell-phone, both switched on. His laptop has only a Wifi and an Ethernet adapter without available network in range. On the contrary, his cell-phone is connected to a HSDPA network and has its own Wifi adapter (but as for the laptop, without access point available). Bob's manager has just sent a new questionnaire; an automatic mail is sent to him with a link to the published questionnaire. In the traditional case, Bob should open the mail, go on the forum and access to the questionnaire on his phone. We suggest that this interaction could be dramatically improved. Let's consider that Bob is deeply focused on his laptop and that his cell-phone lies at some distance of him. Here a simple but still efficient device collaboration rule can take place:

- When receiving a new mail on cell-phone;
- If User is working on superior ergonomics device (Computer, Laptop …) which can be connected with cell-phone;
- Then perform tasks: establish a bridge connection from cell-phone to computer and notify user of the new mail.

Fig 4 represents what resources are necessary to perform the “Bridge Connection” task.

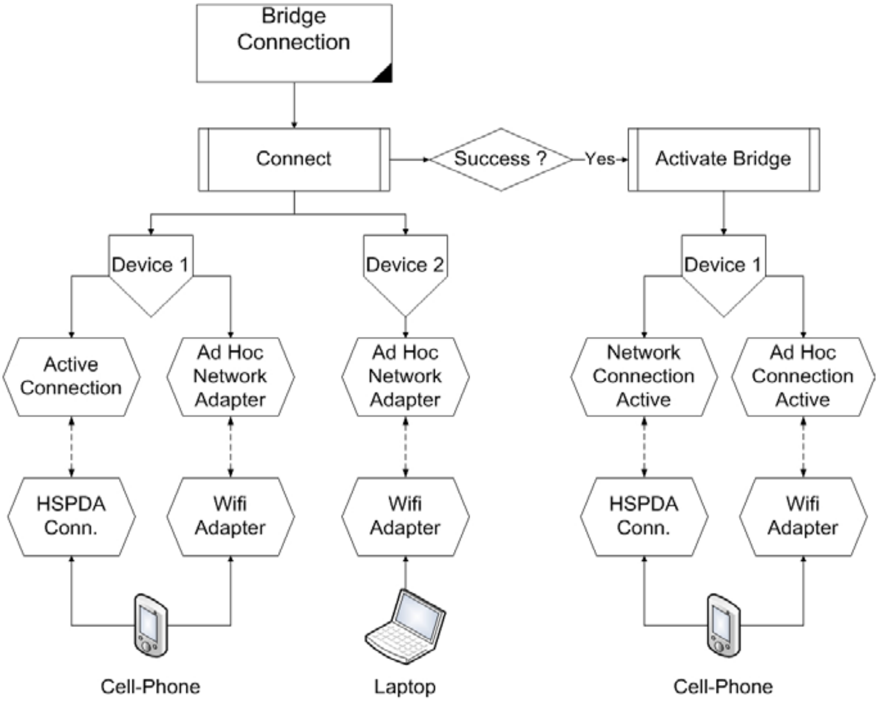


Fig. 4 Device collaboration, connection bridging

As we can see on the previous figure our model serves at representing resources of users’ task to find matching devices but also to determine if those devices can cooperate to supply the desired resources. Indeed to perform the connection bridge between two devices we have defined the required resources: an active connection on the device that have to bridge it; and a network adapter of the same type on both of devices. In our case the available connection can be found on the cell-phone as the HSPDA one while common network is supported by Wifi adapters. Once the cell-phone and laptop are connected the bridge can be activated.

In order to illustrate more precisely the device collaboration rules and their relative mechanisms we can consider a second example. This second use case implies the collaboration of 3 coworkers. Leela, Amy and Philip are members of a team and have to collaborate on a new marketing campaign for the new product of their company. In this perspective they have to perform several tasks together. Let's suppose that they have to make a brainstorming session to design a new advertising board. Amy is working at their main office, but Leela and Philip are not physically present. Leela is working at her home while Philip is in mission in Kenya. In order to be able to work at the same time Amy has sent invitations to Leela and Philip for a virtual Brainstorming with a dedicated software at 3 PM (GMT). In a "device consideration" Amy is working on her usual workstation, Leela has its personal laptop, Philip on his side has a tablet-pc and a smartphone. At 3 Amy has started the server part of the application and has connected her station. At the same time Leela's laptop and Philip's tablet-pc need to connect to the Internet in order to be able to join the Brainstorming platform. To do it they rely on the PCSCW model that should allow their devices to make the right decision. The task associated with the brainstorming activity described with the PCSCW implies several constraints on the resources used by the devices. For the "Connect to Internet" action we've got constraints on several resources. Firstly we've got four constraints on the network connection:

- 1 security constraint: the encryption has to be at least RSA; this constraint has a Very High criticality as the collaboration taking place is close to confidential;
- 1 availability constraint: the average provided bandwidth has to be at least 0,5mbps, this constraint has a High criticality as the application can work with less bandwidth but user's satisfaction and experience may be dramatically lowered by such limitation;
- 1 reliability constraint: the probability to experience network disconnections has to be less than 1 per hour. As this point doesn't completely stop the collaboration it has an Average criticality;
- 1 cost constraint: the price of the connection has to be less than four dollar a minute. As it doesn't obstruct the collaboration this constraint has an Average criticality.

As for the network connection we also have a constraint on the power supply resource:

- 1 availability constraint: the energy supplied has to be sufficient to maintain the connection for three hours in order to have enough time for the brainstorming session. This constraint has a High criticality.

Leela's laptop hasn't many choices and connects itself to the wifi access point of Leela's home's ADSL modem. Philip's situation is totally different. In addition to the tablet-pc, the smartphone and the hotel wifi access point, we've got a description of resources required for the connection to Internet. It also depicts the three possible scenarios to establish the Internet connection:

- Direct connection of the tablet-pc through its satellite network adapter;
- Connection of the tablet with hotel's wifi access point;
- Connection of the tablet with Philip's smartphone with a connection bridge between cellphone's wifi and 3G networks to allow the tablet to acces to Internet.

Each one of these possibilities has advantages and drawbacks:

- Direct connection with satellite network:
  - Advantages: highly secured, only rely on tablet's energy, relatively stable;
  - Drawbacks: slow connection (~0,2mbs) and costly, occasional disconnections;
- Connection to hotel's wifi:
  - Advantages: good bandwidth(~2mbps), free, low energy consuming;
  - Drawbacks: poorly secured (WPA), variable bandwidth, disconnections every fifteen minutes;
- Connection with smartphone:
  - Advantages: as we use ad-hoc wifi the security is up to the two devices and can be relatively good, the average bandwidth is fair (~1 mbps) and the connection is relatively stable;
  - Drawbacks: power supply is limited by smartphone's battery life which is limited to 2.8 hours due to the high energy consumption of the 3G and wifi adapters.

We consider that Philip's tablet has already acquired all these information; he must now find the best solution. This is simply realized by analyzing solutions constraints fulfilments. From the precedent listing of advantages and drawbacks and after having put a score to each constraint in respect with expected and offered values and their criticality we have to come to the conclusion that despite the risk of shortening the brainstorming of some minutes, the smartphone connection option is the one offering the best compromise. Thus the system has been able to automatically find three possible scenarios of connection and has chosen the most adapted solution by a simple comparison between required and available resources and by evaluating the fulfilment of some constraints.

These scenarios show how our model enables the efficient cooperation of surrounding devices to enhance the collaboration of users. Besides, this type of scenario can be further extended to multiple users and devices, creating a real "pervasive and collaborative network".

5 Simulator

For now we have presented our model. However this model has not been tested in real conditions yet. In order to validate it we are currently developing a simulator that will help us in this process. Simulation is a very helpful validation device; it helps understanding the involved collaborations as well as triggers invaluable feedbacks from stakeholders about the elicited requirements. One of our first thoughts was to directly implement a real application to use our model. However, when we studied this project we were rapidly confronted to limitations and difficulties. Among the main difficulties of developing such an application is the necessity to handle various kinds of devices and then a variety of operating systems and environments. Furthermore, even if this problem is not unbearable there is at least another one that can't be simply handled: resources required for each test. Indeed, if you want to validate a model you need to perform batteries of tests, in the case of a real application it can be outrageously costly and time consuming. Moreover, as we're dealing with multiple users and multiple devices interactions, we would have needed many people in different locations with several smart devices. Thus, we quickly found out that our approach was not the good, and that the only way to efficiently validate our model was to develop a simulator. Benefits of such methods are numerous: you don't need to "hire" a bunch of people and send them on the other side of the city in a fast food with free wireless Internet connection, you just have to click on some buttons; you don't need tens of minutes of preparation before each test, you just have to prepare it once and run it as many times as you want; another advantage here is that you can trace every single event of the simulation and analyze it afterwards, which is much more difficult in real situation.

In order to validate the model we have decided to feed our simulator with collaboration scenarios. As we'll see in the rest of this work, scenarios will be composed of an initial state of the world with a given set of agents and by some events that will trigger collaboration rules among devices. As we already said we will trace the execution of the scenario and the different actions made by agents, in order to be able to replay and analyze them. Thus the simulator will be a real laboratory to construct and improve devices collaboration rules.

5.1 Architecture

The architecture of the simulator has been organized in several modules articulated according to presentations, controllers and data. As the presentation part of the simulator isn't really important and can be changed without interfering with the business aspect of the simulation, we will not spend more time on this part. On the opposite, controllers are the heart of the system. They ensure the management of the application in a technical and business perspective. Most important data of the simulation are stored in an ontology and its associated set of semantic rules.

Technically, as we had chosen to use the JADE[1] framework for the multi-agent aspect we naturally decided to develop our simulator in Java. To access, update and manage the ontology we rely on the Protégé[2] Framework developed at the Stanford University that allowed us to write our ontology in OWL[3]. To be completely useful an ontology needs semantic rules; for this part we decided to follow the recommendations of the W3C and use the SWRL[4] language.

In a "business" perspective, the ontology will be used to represent all data of the scenario, that is to say all information about the initial state of the scenario and

[1] JADE : `http://jade.tilab.com/`

[2] Protégé: `http://protege.stanford.edu/`

[3] OWL: `http://www.w3.org/TR/owl-features/`

[4] SWRL: `http://www.w3.org/Submission/SWRL/`

all events that are scheduled to occur in the simulation (all events that do not implies the behavior of devices, such as date, time, spatial consideration, human interactions and events, etc). For the reasoning itself, device collaboration rules are modeled using SWRL Rules. These rules are composed of two parts: the *antecedent* that represent the conditions to be evaluated and the *consequent* representing the result (the implications) of the rule when all conditions are verified. In our cases where they represent devices collaboration rules, the formalism is the following: the antecedent represent the conditions required to activate the consequent. Thus the consequent represent the action to be performed by the agent in term of resources

Fig 5 below depicts the overall architecture of the simulator, which is composed of three main aspects: multi-agent, business and data.

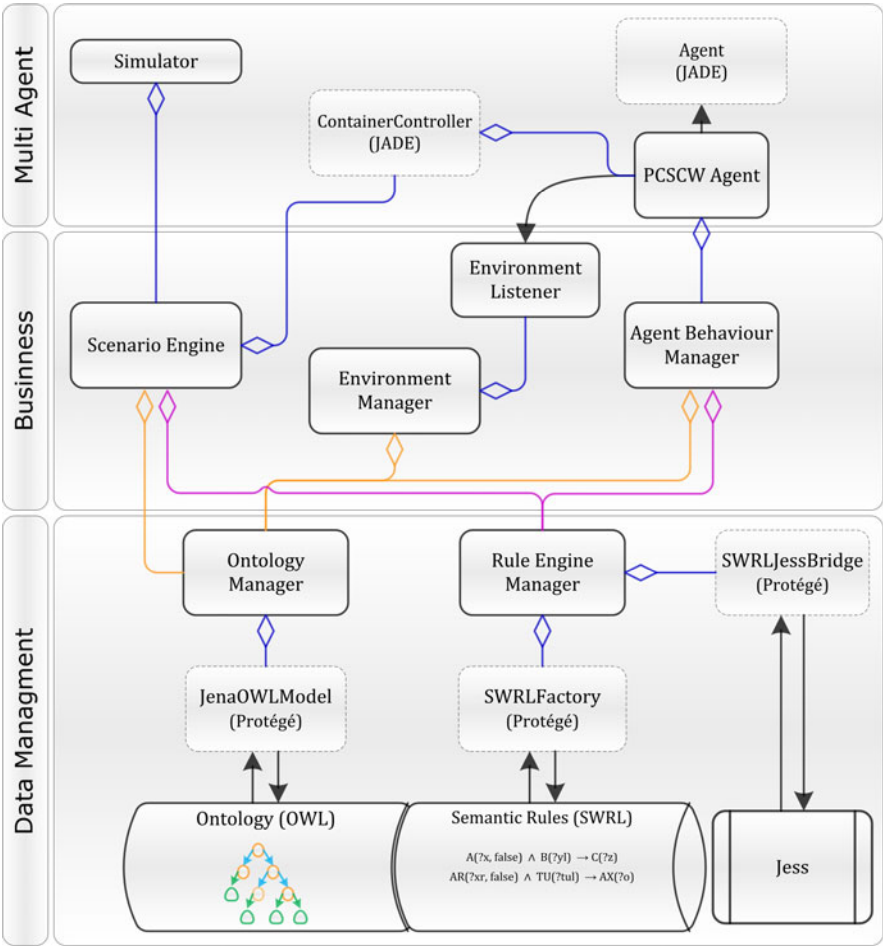


Fig. 5 Simulator's Architecture

On the top of the figure we've got the main object representing the simulator. Its role is to manage available scenarios and provide common features such as loading, saving, help and else. Bellow this main object we've got some vital modules:

Data aspect:

- *Ontology Manager*: this module is a "Facade" as it facilitates the use of the Protégé Framework by providing a set of simple methods specifically developed for the simulator. For instance it allows to efficiently retrieving all information (stored in OWL) about the initial environment of a scenario or about a specific agent.
- *Rule Engine Manager*: the rule engine manager is also a "Facade" but its role is a little different from the *ontology manager*. Indeed, while the *ontology manager* deals with OWL information retrieving and wrapping, the rule engine provides an interface not only for semantic rules (in our case SWRL Rules) but also to control and manage an SWRL Rule Engine which is in charge of running our semantic rules and thus trigger devices collaboration rules.

Business:

- *Scenario Engine*: responsible of the schedule and trigger of events in the simulation, it also has in charge to create the initial environment and manage the *ContainerController* of the JADE Framework that holds agents.
- *Environment Manager*: this module ensures the management of the Environment; it handles environment updates and dispatches environment events to agents.
- *Environment Listener*: this small module is used as an interface between the environment and agents; it provides simple methods to communicate environment's updates to agents and agents' updates to the environment.
- *Agent Behaviour Manager*: this is one of the most important modules of the simulator, it manages how the agent will react according to its context and preserve the coherence of its behaviour. For instance if an agent starts an action that may require several interactions with other agents, it has in charge to "memorize" and make the agent follow the process of the behaviour.

Multi-Agent:

- *PCSW Agent*: the base agent in our simulator, it inherits from the JADE *Agent* class and implements specific features to communicate with the environment and its *Agent Behaviour Manager* described above.
- *Simulator*: as already mentioned it is the main module of the application, managing the load of scenarios and eventually responsible for the management of one or several presentations.

5.2 Simulator's Streams

We have just detailed the technical architecture of our simulator; let's have a look at the way we feed it and how it provides us our useful results as it is presented on Fig 6.

The main input is obviously the scenario; it is composed of an initial environment storing all information about agents and other resources to represent the initial state of the scenario. In addition to this initial environment the scenario stores roles and tasks used by agents. In order to ensure the unfolding of the scenario we've got a set of events representing the different steps of the scenarios. Each event is triggered according to the evolution of the environment and agents. The second input managed by the simulator is device collaboration rules. As we want to make them as generic as possible we don't bound this rules to specific cases and we can then separate them from scenarios. We haven't represented them on the figure but it is obvious that we've got some other technical parameters for the simulation that do not directly interfere with the playing of scenario but with the user interface.

Once a scenario has reached its final step or its time limit it also has produced a set of outputs. The most obvious one is a report of scenario results, showing the execution time, the list of agents involved in the collaboration the last step reached and the state of the environment and agents at the end. We also produce some statistics concerning the scenario such as time to accomplish each task, the number of interactions between human and machines, any indicator that can help us analyze

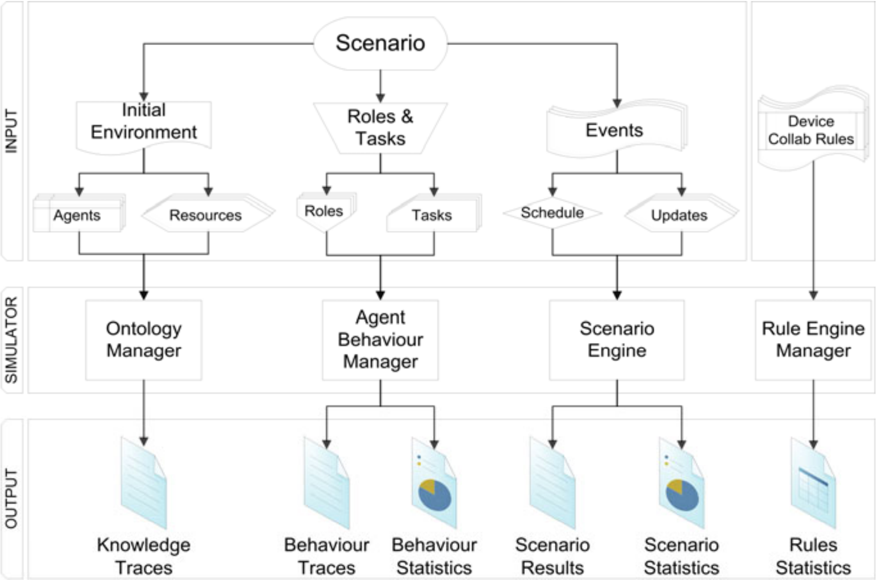


Fig. 6 Simulator's Input and Output

the scenario. Concerning this last point we already have some trails [19] about the kind of indicator that could help us evaluate the collaboration in pervasive computing environments as we'll see in the following section. We also produce other kind of outputs: rules and behaviors statistics allowing us to evaluate the utility of devices collaboration rules. The last type of output consists in traces of behaviors and agents knowledge; combined with the original scenario they can be used to "replay" the past simulation. In addition this kind of output is particularly useful as it can help us finding way of improvement for device collaboration rules.

Let's see how these outputs can serve us to evaluate our work and the collaboration in pervasive computing environments.

6 Evaluation

Until now we have presented our new model for collaborative work in pervasive computing environments, the PCSCW model. In addition we are currently in the process of developing a simulator to validate this model. There is still a point that we haven't considered in this paper: how can we evaluate the model, its benefits and its drawbacks.

Classical computer supported collaborative work evaluation offers a large variety of methods. Thus, many different techniques have been used to evaluate groupware technologies, but for now no consensus has been reached on which methods are appropriate in which context. Existing CSCW evaluation methods can be organized according to two main categories: traditional methods applied to CSCW -the first tries to evaluate CSCW were done using single user methods- and methods especially created for it. Among these two main categories we can point out some similarities: *discount methods* aimed at providing low-cost evaluation, they offer a mean to quickly evaluate a system with limited cost and most of the time with few constraints but they have some limitations such as dissociation from the real work settings, lack of a real theoretical basis, weak coupling with the domain, lack of accuracy, *scenario-based methods* are effective in helping to focus evaluation efforts and in identifying the range of technical, human, organizational, and other contextual factors that impact system success. However, this kind of method is somewhat less useful for identifying the measurable benefits gained from a CSCW implementation due to the potential complexity to determine what part of the system has improved the collaboration. Finally we've got *task model based evaluation* aimed at evaluating the role of a user inside the collaboration, trigger some events at some point of the task, make statistics about the tasks, feed a context manager, but the applicability of this type of formal analysis is limited by the availability of quantitative data concerning the application, which in the case of collaborative software can be complex to collect and even more to interpret correctly.

As our work sticks a little out of the traditional perspective we had to find how we could evaluate our work. Such model cannot be fully evaluated theoretically; indeed to do it efficiently we need to evaluate its use. Then we base our evaluation on the analysis of the efficiency of the representation coupled to devices collaboration rules in the improvement of collaboration between humans.

Thus, one of the first things we needed to consider to conduct an evaluation of this model is how can we quantify benefits and drawbacks, how can they be measured, what kind of evaluation indicators can be used. To find those indicators we considered the real purpose of our model: simplifying the interactions between humans by automating some of the interactions that can take place between smart devices. From this perspective we can point out some concrete indicators:

- *Number of devices interactions to complete a task*: this measure has to be interpreted according to the objective of the collaboration, indeed users may want to limit exchanges between devices for cost or security reasons for example. Thus, a high number of interactions may mean that devices have well adapted their behaviour to help users, but also that they have overexposed users' documents.
- *Number of human interactions to complete a task*: this simple measure represents the number of human interactions (with devices of with humans) that has been required to perform a given task. For this indicator, fewer interactions often mean a simpler and more efficient collaboration.
- *Number of interacting humans / total number of humans*: this measure can indicate the complexity of the collaboration, the more humans there is to collaborate the more complex the collaboration will be. Then if there are fewer humans involved to perform a task it implies that our model facilitates the collaboration.
- *Time to complete a task*: this simple measure can have an important impact on the relevance of a device collaboration rule, indeed, even if a rule efficiently reduce the number of humans and devices interactions, it may not be usable if it dramatically increase the required time.

In order to facilitate the validation and the evaluation of our model we have designed our simulator to be able to simply handle this kind of measure. Thus, once completed we will have the opportunity to measure and compare the efficiency of collaboration with and without our model. This will also help us to improve our devices collaboration rules by providing us useful indicators.

7 Discussion

The model we propose does not come out of nowhere, it relies on robust researches that inspired us and guided us to develop it. As it has been intensively mentioned, our model is based on the notion of roles, for people and for devices. In a moral consideration, the representation of roles is not a substitution of the representation of a person, it is only a part of a person, otherwise one can quickly come to the conclusion that only roles matters and peoples don't. But from a model perspective taking into account the role as a variable can help to apprehend the complexity of a pervasive environment. In such context, roles or resources can vary depending on spatial, temporal or collaborative constraints. Having a model in which the "efficiency" of the collaboration can be estimated may be used for designing purpose.

The PCSCW model is designed to facilitate the collaboration of users by making devices cooperate. In its nature this model could be considered as a meta-model as it tells how some sub models can be used and combined to improve the collaboration of users. Even though we can take out some benefits and drawbacks of our model put in regard with previous ones.

The most obvious benefit or our model compared with others is the fact that it natively considers the distribution of resources and the possibility to use them all at once. Indeed most of traditional collaboration models, based on tasks, roles or even more advanced collaboration awareness models focus on the way to keep users and their devices aware of the collaboration. We think the PCSCW model is going further in this direction by using collaboration awareness to enable the "collaboration intelligence" of devices and develop their proactive behaviors. Another noticeable and valuable benefit of our approach is the possibility to precisely monitor the current state of the collaboration. Indeed, as we have to depict each task and their related actions it helps channeling the collaboration awareness.

However we know that there are some drawbacks in our approach, the main one is probably the high level of description required by devices to adapt their behavior. Thus if we only need to have an overview of the collaborative activity some models (such as [12]) can propose faster, but less accurate, solutions than ours.

Indeed, our approach can seem very descriptive, detailed and requiring great efforts to be used. But we want to take an advantage from this issue. In fact, all awareness mechanisms do not require the same level of description. For some of them, only the top levels are relevant. This is why we argue our model is able to describe and reason on different granularity levels, from a simple description of devices until a fine description of each object manipulated by an application on a virtualized operating system. Thus, we can say that our model naturally supports the scalability of awareness mechanisms by its adaptability to the description of resources. This scalability can even bring an abstraction capacity by allowing designers to represent high-level information and reason on it. Besides, this scalability advantage is twofold, it allows the description of resources with various granularity, but it also offers the possibility to reason with few context information and then when computing resources are limited or information are hard to obtain.

This work relies on two main aspects: the representation of required and available resources and the description of device collaboration rules. Still we know our work has its own disadvantages. One of the mains is the need to create these rules. Indeed to adapt to a specific context it requires having a more or less generic set of rules. Even if this particular point can seem annoying it can be a source of improvement. Despite rules have to be written before the use of the model, they can also be derived from user's activity, preferences and constraints dynamically. As for now we don't have dig deeper in this way as it was not our main focus, but in future works we'll need to explore potential way to automatically generate and adapt rules.

We think our model is a good basis to develop interactions between smart devices. Besides the collaboration supported is threefold: collaboration between users, collaboration between devices and collaboration between users and devices.

Still, we know that our model certainly needs some improvements, and that without implementation and in-depth evaluation it is only theory. In this perspective the simulator we're currently developing will provide use a useful tool to validate and evaluate the efficiency of the model. Even if it is an essential tool for the evaluation process it is not sufficient for a real deployment. Indeed, if we refer to [14] and [15], the evaluation of a CSCW system has to be organized in three phases: laboratory evaluation without users to detect obvious problems, laboratory evaluation with a part of user's context and evaluation in real conditions to validate the scalability of the system and its deployed efficiency. Hence the next step for us after the development of the simulator and the laboratory evaluation will be to develop a prototype of the final application and make it evaluate by real users. There's no doubt this prototype will be very helpful to design new collaboration rules as users' feedbacks will go along. Combined together the simulator and the prototype will give us an efficient evaluation framework to create and improve devices collaboration rules in order to facilitate even more the collaboration between users.

References

1. Zhu, H., Alkins, R.: A Tool for Role-Based Chatting. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 3795–3800 (2007)
2. Zhu, H.: From WYSIWIS to WISINWIS: Role-Based Collaboration. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 5441–5446 (2004)
3. Zhu, H., Tang, Y.: A Role-Based Hierarchical Group Collaborative Awareness Model and its Implementation. In: IEEE Information, Technologies and applications in Education, pp. 12–16 (2007)
4. Molina, A.I., Redondo, M.A., Ortega, M., Hoppe, U.: CIAM: A Methodology for the Development of Groupware User Interfaces. Journal of Universal Computer Science 14, 1435–1446 (2008)
5. Ahn, G.J., Zhang, L., Shin, D., Chu, B.: Authorization Management for Role-Based Collaboration. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 4128–4134 (2003)
6. Barbuceanu, M., Gray, M., Mankovski, S.: Coordinating with Obligations. In: Agents 1998, Minneapolis, pp. 62–69 (May 1998)
7. Edwards, W.K.: Policies and Roles in Collaborative Applications. In: Proceedings ACM Conference Computer Supported Cooperative Work, Cambridge, MA, pp. 11–20 (1996)
8. Hamadache, K., Manson, P., Lancieri, L.: Pervasive services, brainstorming in situation of mobility. In: 3rd International Conference on Pervasive Computing and Applications, Alexandria, Egypt, pp. 709–714 (2008)
9. Penichet, V., Lozano, M., Gallud, J.A., Tesoriero, R.: Task Modelling for Collaborative Systems. In: Winckler, M., Johnson, H., Palanque, P. (eds.) TAMODIA 2007. LNCS, vol. 4849, pp. 287–292. Springer, Heidelberg (2007)
10. Yan, L., Zeng, J.: A Task-Based Group Awareness Model. In: Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, May 26-28, vol. 2, pp. 90–94 (2004)
11. Vellis, G.: Model-based development of synchronous collaborative user interfaces. In: Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Pittsburgh, PA, USA (2009)

12. Rodden, T.: Populating the application: A model of awareness for cooperative applications. In: Proceedings of the ACM CSCW 1996 Conference on Computer Supported Cooperative Work, pp. 87–96. ACM Press, Boston (1996)
13. Zhu, H., Seguin, P.: The Role Transition Mechanisms in Role-based Collaborative Systems. In: Canadian Conference on Electrical and Computer Engineering, pp. 1359–1362 (2005)
14. Herskovic, V., Pino, J.A., Ochoa, S.F., Antunes, P.: Evaluation Methods for Groupware Systems. In: Haake, J.M., Ochoa, S.F., Cechich, A. (eds.) CRIWG 2007. LNCS, vol. 4715, pp. 328–336. Springer, Heidelberg (2007)
15. Hamadache, K., Lancieri, L.: Strategies and Taxonomy, Tailoring your CSCW Evaluation. In: Baloian, N., Fonseca, B. (eds.) CRIWG 2009. LNCS, vol. 5784, pp. 206–221. Springer, Heidelberg (2009)
16. Mori, G., Paternò, F., Santoro, C.: CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Trans. Softw. Eng. 28(8), 797–813 (2002)
17. Drury, J., Williams, M.G.: A framework for role-based specification and evaluation of awareness support in synchronous collaborative applications. In: Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies, pp. 12–17. IEEE Press, Pittsburgh (2002)
18. Gutwin, C., Greenberg, S.: A descriptive framework of work space awareness for real-time groupware. In: Computer-Supported Cooperative Work, pp. 411–446 (2002)
19. Hamadache, K., Lancieri, L.: Dealing with Device Collaboration Rules for the PCSCW Model. In: Kolfschoten, G., Herrmann, T., Lukosch, S. (eds.) CRIWG 2010. LNCS, vol. 6257, pp. 233–248. Springer, Heidelberg (2010)
20. Linxia, Y.: Realization of task-based group awareness model. In: International Conference on Computational Intelligence and Software Engineering, pp. 1–3 (2009)

Evolving Schedules of Independent Tasks by Differential Evolution

Pavel Krömer, Václav Snášel, Jan Platoš, Ajith Abraham, and Hesam Ezakian

Abstract. Scheduling is one of the core steps to efficiently exploit the capabilities of heterogeneous distributed computing systems and it is also an appealing NP-complete problem. There is a number of heuristic and meta-heuristic algorithms that were tailored to deal with scheduling of independent jobs. In this paper we investigate the efficiency of differential evolution for the scheduling problem and compare it with existing approaches. The analysis shows that the differential evolution is a promising method that can compete with well-established scheduling algorithms.

1 Introduction

Grid computing and distributed computing, dealing with large scale and complex computing problems, is a hot topic in the computer science and research. Mixed-machine heterogeneous computing (HC) environments utilize a distributed suite of different machines connected via the computer network to perform different computationally intensive applications that have diverse requirements [1, 2]. Miscellaneous resources ought to be orchestrated to perform a number of tasks in parallel, or to solve complex tasks atomized to a variety of independent subtasks [3]. Proper

Pavel Krömer · Václav Snášel · Jan Platoš
Department of Computer Science, VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
e-mail: {pavel.kromer,vaclav.snasel,jan.platos}@vsb.cz

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs), Washington 98071, USA
e-mail: ajith.abraham@ieee.org

Hesam Izakian
Department of Computer Engineering, University of Isfahan,
Hezar Jerib Avenue, Isfahan, Iran
e-mail: hesam.izakian@gmail.com

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 79–94.
springerlink.com

scheduling of the tasks on available resources is one of the main challenges of a mixed-machine HC environment.

To exploit the different capabilities of a suite of heterogeneous resources, a resource management system (RMS) allocates the resources to the tasks and the tasks are ordered for execution on the resources. At a time interval in HC environment, number of tasks are received by the RMS. Task scheduling is mapping of a set of tasks to a set of resources to efficiently exploit the capabilities of such.

It is known that an optimal mapping of computational tasks to available machines in a HC suite is a NP-complete problem [4] and as such, it is a subject to various heuristic and meta-heuristic algorithms. The heuristics applied to the task scheduling problem include min-min heuristic, max-min heuristic, longest job to fastest resource- shortest job to fastest resource heuristic, sufferage heuristic, work queue heuristic and others [2, 5, 6]. The meta-heuristics applied to the task scheduling problem include hybrid ant colony optimization [7], simulated annealing [8] and genetic algorithms [9, 10, 11]. The meta-heuristic algorithms usually operate with a population of prospective problem solutions - task schedules - that are evolved (optimized) in order to obtain an improved schedule which is optimized according to some criteria.

In this paper, we apply a powerful populational meta-heuristic algorithm - the differential evolution - to the task scheduling problem. Moreover, to improve the efficiency of the general meta-heuristic solver, several widely used heuristic algorithms for scheduling in the HC environments were used to improve the initial population for differential evolution.

2 Heuristic Algorithms for Mapping Tasks in HC Environment

There is a number of heuristic algorithms designed to schedule independent tasks in heterogeneous computing environments. Each algorithm exploits a heuristic based on certain intuition that helps mapping tasks to machines so that selected objective is optimized. Unfortunately, different heuristics perform under various circumstances differently [12, 6].

In [12] was presented an analysis of 16 heuristic algorithms for the task scheduling problem. The authors have shown that different scheduling heuristics perform differently for different expected time to compute (ETC) matrices. The ETC matrices were generated with various values of task heterogeneity V_{task} and machine heterogeneity $V_{machine}$. A series of computational experiments has shown that the result of different heuristic algorithms depends on the values of V_{task} and $V_{machine}$. For the ETC matrices with both, high V_{task} and high $V_{machine}$, the Heaviest Task First (HTF) [12] heuristic performed better than other heuristic algorithms. For the ETC matrices with either medium V_{task} or medium $V_{machine}$ obtained best results heuristic algorithm called Task Partitioning Min (TPMin) introduced in the same study. The ETC matrices with low V_{task} or low $V_{machine}$ were solved by the heuristic algorithm Task Partitioning Std (TPStd) with greatest success. The details on TPMin, TPStd and the description of the experiments can be found in [12]. In short, to obtain best schedules, the heuristic algorithm had to be chosen according to the ETC matrix properties V_{task} and $V_{machine}$. Such an observation encourages the investigation of adaptive, meta-heuristic, approaches to the scheduling problem that are supposed to yield good performance for ETC matrices with various values of V_{task} and $V_{machine}$.

Efficient heuristic algorithms for scheduling in HC environments include [5, 6, 2]:

- Min-min heuristic that prioritizes tasks that can be completed earliest.
- Max-min heuristic that prioritizes tasks with the maximum earliest completion time. It aims to overlap long-running tasks with short-running ones.
- Sufferage heuristic that is based on the idea that better mappings can be generated by assigning a machine to a task that would suffer most in terms of expected completion time if that particular machine is not assigned to it.
- Min-max heuristic that combines two metrics, the minimum execution time and the minimum completion time. It aims to assign the task to a machine that can handle it with lower execution time in comparison with other machines.

We provide brief description of discussed scheduling heuristics.

2.1 Min-min Heuristics

Min-min heuristic uses minimum completion time (MCT) as a metric, so that the task which can be completed the earliest is given priority. This heuristic begins with the set U of all unmapped tasks. Then the set of minimum completion times $M = \{min(completiontime(T_i, M_j)), i \in [1,n], j \in [1,m]\}$ is found M consists of one entry for each unmapped task.

Next, the task with the overall minimum completion time from M is selected and assigned to the corresponding machine and the workload of the selected machine will be updated. And finally the newly mapped task is removed from U and the process repeats until all tasks are mapped (i.e. U is empty)[2, 13].

2.2 Max-min Heuristic

The Max-min heuristic is very similar to min-min and its metric is MCT as well. It starts with the set of all unmapped tasks U . Then, the set of minimum completion times $M = \{min(completiontime(T_i, M_j)), i \in [1,n], j \in [1,m]\}$ is found. Next, the task with the overall maximum completion time from M is selected and assigned to the corresponding machine and the workload of the selected machine is be updated. Finally, the newly mapped task is removed from U and the process repeats until all tasks are mapped [2, 13].

2.3 Sufferage Heuristic

In this heuristic, the fastest and second fastest completion times are found for each task in the first step. The difference between these two values is called the sufferage value. In the second step, the task with the maximum sufferage value is assigned to the corresponding machine with minimum completion time. The Sufferage heuristic is based on the idea that better mappings can be generated by assigning a machine to a task that would ”suffer” most in terms of expected completion time if that particular machine is not assigned to it [14].

2.4 Min-max Heuristic

This heuristic is composed of two steps for mapping each task [6]. It uses the minimum completion time in the first step and the minimum execution time in the second step as a metric. In the first step, this heuristic begins with the set U of all unmapped tasks. Then the set of minimum completion times, $M = \{min(completiontime(T_i, M_j)), i \in [1,n], j \in [1,m]\}$, is found in the same manner as in min-min heuristic.

In the second step, the task whose minimum execution time (time for executing task on the fastest machine) divided by its execution time on the selected machine (from first step) has the maximum value, is selected for mapping. The intuition behind this heuristic is that it selects pairs of machines and tasks so the machine can execute assigned task effectively with a lower execution time in comparison to other machines.

2.5 Scheduling Heuristics – a Summary

From the optimization point of view, each heuristic algorithm represents a strategy that finds a local optimum among all possible schedules. Despite the fact that the heuristic methods obtain quickly good suboptimal results, its inconsistent behavior (i.e. different efficiency of the algorithms for ETC matrices with different values of $V_{machine}$ and V_{task}) encourages the research of global optimization methods for scheduling in HC environments. In this paper, we investigate scheduling of independent tasks as a combinatorial optimization problem and we use differential evolution to find good schedules.

3 Differential Evolution

Differential evolution (DE) is a reliable, versatile and easy to use stochastic evolutionary optimization algorithm [15]. DE is a population-based optimizer that evolves real encoded vectors representing the solutions to given problem. The real-valued nature of population vectors differentiates the DE notably from GAs that were designed to evolve solution encoded into binary or finite alphabets.

The DE starts with an initial population of N real-valued vectors. The vectors are initialized with real values either randomly or so, that they are evenly spread over the problem domain. The latter initialization usually leads to better results of the optimization process [15].

During the optimization, DE generates new vectors that are perturbations of existing population vectors. The algorithm perturbs vectors with the scaled difference of two randomly selected population vectors and adds the scaled random vector difference to a third randomly selected population vector to produce so called trial vector. The trial vector competes with a member of the current population with the same index. If the trial vector represents a better solution than the population vector, it takes its place in the population [15].

Differential evolution is parameterized by two parameters [15]. Scale factor $F \in (0, 1+)$ controls the rate at which the population evolves and the crossover probability $C \in [0, 1]$ determines the ratio of bits that are transferred to the trial vector from its opponent. The number of vectors in the population is also an important parameter of the population. The outline of DE is shown in Figure 1.

```
Initialize the population $P$ consisting of $M$ vectors
Evaluate an objective function ranking the vectors in the population
while Termination criteria not satisfied do
    for $i \in \{1, \ldots, M\}$ do
        Create trial vector $v_t^i = v_r^1 + F(v_r^2 - v_r^3)$, where $F \in [0, 1]$ is a parameter
        and $v_r^1$, $v_r^2$ and $v_r^3$ are three random vectors from the population $P$. This
        step is in DE called mutation.
        Validate the range of coordinates of $v_t^i$. Optionally adjust coordinates of $v_t^i$
        so, that $v_t^i$ is valid solution to given problem.
        Perform uniform crossover. Select randomly one point (coordinate) $l$ in $v_t^i$.
        With probability $1 - C$ let $v_t^i[m] = v^i[m]$ for each $m \in \{1, \ldots, N\}$ such that
        $m \neq l$
        Evaluate the trial vector. If the trial vector $v_t^i$ represent a better solution
        than population vector $v^i$, replace $v^i$ in $P$ by $v_t^i$
    end
end
```

Fig. 1 A summary of Differential Evolution

There are more variants of differential evolution. They differ mostly in the way new vectors are generated.

Differential evolution represents an alternative to the more traditional concept of genetic algorithms. As well as genetic algorithms, it represents a highly parallel population based stochastic search metaheuristic. In contrast to GA, differential evolution uses real encoding of chromosomes and different operations to maintain and evolve the population. It results in different search strategy and different directions found by DE when crawling the fitness landscape of a particular problem domain.

4 Differential Evolution for Scheduling Optimization

In general, a combinatorial optimization problem is a problem whose solution is discrete or can be reduced to a discrete one. More formally, a combinatorial optimization (CO) problem $\mathrm{P} = \{I, \{sol(i)\}_{i \in I}, m\}$ can be defined as a minimization or maximization problem that consists of a set of problem instances I, a set of feasible solutions $sol(i)$ for every instance $i \in I$ and a function $m : \{(i,q)|i \in I, q \in sol(i)\} \rightarrow \mathrm{Q}_+$, where Q_+ is the set of positive rational numbers and $m(i,q)$ is the value of solution q for the problem instance i [16]. An optimal solution to an instance of a combinatorial optimization problem is such solution that has maximum (or minimum) value among all other solutions. Famous combinatorial optimization problems include among others the traveling salesman problem, the knapsack problem, and the linear ordering problem [16]. For the job scheduling problem, the set of feasible solutions corresponds to the set of all possible schedules and an optimal schedule is the schedule with lowest value of selected fitness function.

A HC environment is composed of computing resources. These resources can be a single PC, a cluster of workstations or a supercomputer. Let $T = \{T_1, T_2, \ldots, T_n\}$ denote a set of tasks that is at a specific time interval submitted to the RMS. Assume that the tasks are independent of each other with no inter-task data dependencies and preemption is not allowed, i.e. the tasks cannot change the resource they have been assigned to. Also assume at the time of receiving these tasks by RMS, m machines $M = \{M_1, M_2, \ldots, M_m\}$ are within the HC environment. For our purpose, scheduling is done on machine level and it is assumed that each machine uses First-Come, First-Served (FCFS) method for performing the received tasks. We assume that each machine in the HC environment can estimate how much time is required to perform each task. In [2], the expected time to compute (ETC) matrix is used to estimate the required time for executing a task in a machine. The ETC matrix is a $n \times m$ matrix in which n is the number of tasks and m is the number of machines. One row of the ETC matrix contains the estimated execution time for a given task on each machine.

Similarly, one column of the ETC matrix consists of the estimated execution time of a given machine for each task. Thus, for an arbitrary task T_j and an arbitrary machine M_i , $[ETC]_{j,i}$ is the estimated execution time of T_j on M_i. In the ETC model, we take the usual assumption that we know the computing capacity of each resource, an estimation or prediction of the computational needs of each job, and the load of prior work of each resource.

The objectives to optimize during the task mapping are makespan and flowtime. Optimum makespan (metatask execution time) and flowtime of a set of jobs can be defined as:

$$makespan = \min_{S \in Sched} \{ \max_{j \in Jobs} F_j \} \tag{1}$$

$$flowtime = \min_{S \in Sched} \{ \sum_{j \in Jobs} F_j \} \tag{2}$$

where $Sched$ is the set of all possible schedules, $Jobs$ stands for the set of all jobs and F_j represents the time in which job j finalizes. Assume that $[C]_{j,i}$ $(j = 1, 2, \ldots, n,$

$i = 1,2,\ldots,m)$ is the completion time for performing j-th task in i-th machine and W_i $(i = 1,2,\ldots,m)$ is the previous workload of M_i, then $\sum(C_i + W_i)$ is the time required for M_i to complete the tasks included in it. According to the aforementioned definition, makespan and flowtime can be evaluated using Eq. (3) and Eq. (4) respectively.

$$makespan = \min_{i \in \{1,\ldots,m\}} \{\sum C_i + W_i\} \tag{3}$$

$$flowtime = \sum_{i=1}^{m} C_i \tag{4}$$

Minimizing makespan aims to execute the whole metatask as fast as possible while minimizing flowtime aims to utilize the computing environment efficiently.

4.1 Schedule Encoding

A schedule of n independent tasks executed on m machines can be naturally expressed as a string of n integers $S = (s_1, s_2, \ldots, s_n)$ that are subject to $s_i \in 1,\ldots,m$. The value at i-the position in S represents the machine on which is the i-the job scheduled in schedule S. This encoding is in the literature known as direct encoding [10].

Because the differential evolution encodes candidate solutions as real vectors, real coordinates have to be used instead of discrete machine numbers. The real-encoded DE vector is translated to schedule representation by truncation of its elements.

4.2 Schedule Evaluation

Assume a schedule S from the set of all possible schedules $Sched$. For the purpose of the differential evolution, we define a fitness function $fit(S) : Sched \rightarrow \mathbb{R}$ that evaluates each schedule:

$$fit(S) = \lambda \cdot makespan(S) + (1 - \lambda) \cdot \frac{flowtime(S)}{m} \tag{5}$$

The function $fit(S)$ is a sum of two objectives, the makespan of the schedule S and the flowtime of the schedule S divided by the number of machines m to keep both objectives in approximately the same magnitude. The influence of makespan and flowtime in $fit(S)$ is parameterized by the variable weight λ. The same schedule evaluation was used also in [10].

Flowtime and makespan are computed using a binary schedule matrix $B(S) : Sched \rightarrow \{0,1\}^2$ which is constructed as follows: for a $n \times m$ ETC matrix that describes estimated execution times of n jobs on m machines, the $m \times n$ schedule matrix $B(S)$ has in i-th row and j-th column 1 iff the task j is scheduled for execution on machine i. Otherwise, $B(S)_{i,j}$ is equal to 0. Then $flowtime(S) : Sched \rightarrow \mathbb{R}$ and $makespan(S) : Sched \rightarrow \mathbb{R}$ can be defined with the help of matrix multiplication as:

$$makespan(S) = \sum [B(S) \cdot ETC]_{j,j} \tag{6}$$

$$flowtime(S) = \max_{j \in \{1,\ldots,m\}} \sum [B(S) \cdot ETC]_{j,j} \tag{7}$$

Less formally, makespan equals to the sum of all elements on the main diagonal of $B(S) \cdot ETC$ and flowtime equals to maximal value on the main diagonal on $B(S) \cdot ETC$. An example of makespan and flowtime computation for a particular schedule of 3 tasks on 2 machines is shown in ([8]).

$$ETC = \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}, \qquad B(S) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$B(S) \cdot ETC = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 1 & 2 \end{pmatrix} \tag{8}$$

4.3 Related Work: Genetic Algorithms for Job Scheduling

Many nature insired meta-heuristic algorithms were suggested for the task scheduling problem [11, 10]. Among others, simmulated annealing, genetic algorithms, ant colony optimization and particle swarm optimization were used to find optimal schedules for the tasks in computational grids.

Genetic algorithms are a meta-heuristic optimization strategy that is perhaps closest to the differential evolution. In the following, we shortly summarize the differences and similarities between recent approaches to task scheduling based on the genetic algorithms and our implementation of the differential evolution for the task scheduling problem.

4.3.1 Encoding

In [10], two encoding schemes of the task schedules were considered. The direct representation encodes each schedule into a string of n integers $S = (s_1, s_2, \ldots, s_n)$. For the genetic algorithms, S can be used as a chromosome directly, whereas for DE, S can contain real values. It is translated to the schedule by concatenation of the elements of S.

The alternative permutation-based representation is more complex. For each machine m_i, the sequence S_i of jobs assigned to it is constructed. Next, the sequences S_i are concatenated. The resulting vector is a permutation of the jobs assigned to the machines. Such a representation requires maintaining additional information on the number of jobs assigned to each machine.

The direct representation is an intuitive and simple encoding of schedules. Moreover, there is a simple transformation from the direct representation to the permutation-based representation and vice-versa. Therefore, we have used only direct representation modified to suit DE in this study.

4.3.2 Fitness Function

The fitness function for the task scheduling problem in GA was based on the optimization of makespan and flowtime. The two criteria, establishing a multi-objective optimization problem, were in [10] merged using either a hierarchical or a simultaneous approach.

In the hierarchical approach, the criteria were prioritized (makespan received higher priority than flowtime) and in the optimization, criterion with higher importance was not allowed to vary while optimizing according to the criterion with lower importance.

In the simultaneous approach, the schedules were optimized according to both criteria at the same time and a weighted sum as in our fitness function (5) was used to create one scalar fitness value.

5 Experiments

We have implemented differential evolution for scheduling of independent tasks in heterogeneous independent environments. The differential evolution algorithm was implemented in its classic variant referred to as $DE/rand/1/bin$ [15]. To evaluate the performance of the DE for minimizing the makespan and flowtime, we have used the benchmark proposed in [2]. The simulation model is based on the ETC matrix for 512 jobs and 16 machines. The instances of the benchmark are classified into 12 different types of ETC matrices according to the following properties [2]:

- *task heterogeneity* – V_{task} represents the amount of variance among the execution times of tasks for a given machine
- *machine heterogeneity* – $V_{machine}$ represents the variation among the execution times for a given task across all the machines
- *consistency* – an ETC matrix is said to be consistent whenever a machine M_j executes any task T_i faster than machine M_k; in this case, machine M_j executes all tasks faster than machine M_k
- *inconsistency* – machine M_j may be faster than machine M_k for some tasks and slower for others

The DE algorithm was used with the parameters summarized in the Table 1. The parameters were set after brief initial tuning. The factor λ was set to 0.5 to have equal contribution of makespan and mean flowtime to the fitness value.

The experiments were conducted with two different settings for initial population. In the first case, whole initial population was generated randomly. In the second case, the initial population contained some vectors obtained by scheduling heuristics.

Table 1 A summary of DE parameters

Parameter	Value
Population size	20
Terminating generation	100000
Probability of crossover	$C = 0.9$
Scaling factor	$F = 0.1$
Makespan / flowtime ratio	$\lambda = 0.5$

5.1 Experiments with Random Initial Population

Each ETC matrix was named using the pattern $x-y-z$, where x describes task heterogeneity (*h*igh or *l*ow), y describes machine heterogeneity (*h*igh or *l*ow) and z describes the type of consistency (*i*ncosnsistent, *c*onsistent or *s*emiconsistent).

The Table 2 and the Table 3 show makespan and flowtime obtained by max-min heuristic, sufferage heuristic, min-min heuristic, and min-max heuristic. It illustrates that the min-max is the best heuristics to obtain optimal makespan for majority of the ETC matrices and min-min is the best heuristics to obtain optimal flowtime for the majority of the ETC matrices.

The Table 4a and the Table 4b show makespan and flowtime of schedules obtained by the differential evolution algorithm with random initial population.

As apparent from the tables, DE with random initial population cannot compete with domain specific heuristics when optimizing makespan. It ranks fourth and its results are usually better than max-min heuristics, but worse than sufferage heuristics, min-min heuristics and min-max heuristics.

Table 2 Makespan obtained by heuristic algorithms

ETC	max-min	sufferage	min-min	min-max
l-l-c	6753	5461	5468	**5310**
l-l-s	5947	3443	3599	**3327**
l-l-i	4998	2577	2734	**2523**
l-h-c	400222	333413	279651	**273467**
l-h-s	314048	163846	157307	**146953**
l-h-i	232419	121738	113944	**102543**
h-l-c	203684	170663	164490	**164134**
h-l-s	169782	105661	106322	**103321**
h-l-i	153992	**77753**	82936	77873
h-h-c	11637786	9228550	8145395	**7878374**
h-h-s	9097358	4922677	4701249	**4368071**
h-h-i	7016532	3366693	3573987	**2989993**

Table 3 Flowtime obtained by heuristic algorithms

ETC	max-min	sufferage	min-min	min-max
l-l-c	108014	86643	**80354**	84717
l-l-s	95091	54075	**51399**	52935
l-l-i	79882	40235	**39605**	39679
l-h-c	6400684	5271246	**3918515**	4357089
l-h-s	5017831	2568300	**2118116**	2323396
l-h-i	3710963	1641220	**1577886**	1589574
h-l-c	3257403	2693264	**2480404**	2613333
h-l-s	2714227	1657537	**1565877**	1640408
h-l-i	2462485	1230495	1214038	**1205625**
h-h-c	185988129	145482572	**115162284**	125659590
h-h-s	145337260	76238739	**63516912**	69472441
h-h-i	112145666	47237165	**45696141**	46118709

Table 4 Makespan and flowtime obtained by DE with random initial population

(a) Makespan obtained by DE with random initial population.

ETC	DE_{best}	DE_{avg}
l-l-c	7151	7303.2
l-l-s	4479	4582.2
l-l-i	3127	3203
l-h-c	451815	457741
l-h-s	212828	220334
l-h-i	141635	152186
h-l-c	212175	220142.2
h-l-s	141176	142405.2
h-l-i	99413	100307
h-h-c	13325802	13595908
h-h-s	6138124	6545734
h-h-i	4418167	454678

(b) Flowtime obtained by DE with random initial population.

ETC	DE_{best}	DE_{avg}
l-l-c	85422	891272.4
l-l-s	53675	53964.4
l-l-i	43941	44846.2
l-h-c	3783520	3788428
l-h-s	2277816	2383501
l-h-i	1890529	1935355.4
h-l-c	2699241	2765402.2
h-l-s	1597594	1625219.6
h-l-i	1359241	1380342
h-h-c	100921177	104753227
h-h-s	67874790	70281581
h-h-i	57808847	58216428

The Table 3 and the Table 4b show the flowtime of optimized schedules. In this case, the DE reached the best value for two of experimental matrices (l-h-c and h-h-c). Also in the other cases, DE delivered quite competitive results. Obviously, used setting of scheduling DE suited better to the optimization of flowtime.

5.2 Experiments with Optimized Initial Population

In the second set of experiments, the initial population of the DE was upgraded with vectors obtained by scheduling heuristics. Max-min heuristic, sufferage heuristic, min-min heuristic, and min-max heuristic were used to generate four vectors that were included in the initial population. Those vectors were superior to the remaining members of the initial population in terms of makespan and flowtime. The factor *lambda* was set to 0.9 in order to preserve the suboptimal makespan from the initial population because initial experiment showed the tendency to improve flowtime at the expense of good initial makespan. The results of the second round of the DE optimization experiments are summarized in the Table 5b and in the Table 5a respectively.

Table 5 Makespan and flowtime obtained by DE with upgraded initial population

(a) Makespan obtained by DE with upgraded initial population.

ETC	DE_{best}	DE_{avg}
l-l-c	5250	5271
l-l-s	3326	3326.8
l-l-i	2498	2502.2
l-h-c	267773	270912.4
l-h-s	146125	146759.4
l-h-i	100904	101254.6
h-l-c	159770	161262.2
h-l-s	101824	102440.2
h-l-i	76096	76297.4
h-h-c	7775829	7856042.8
h-h-s	4368071	4372414.6
h-h-i	2922633	2953782.6

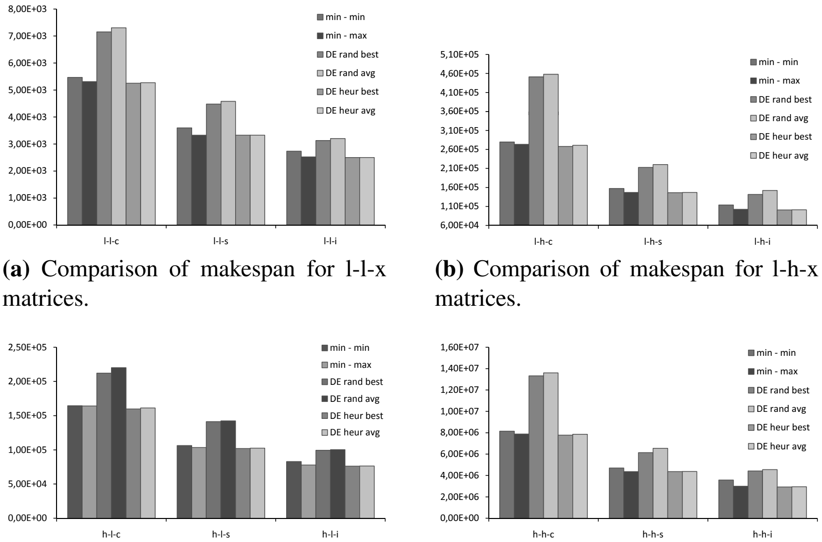
(b) Flowtime obtained by DE with upgraded initial population.

ETC	DE_{best}	DE_{avg}
l-l-c	79580	80785.4
l-l-s	52729	52754.8
l-l-i	39674	39724.6
l-h-c	3829129	3983780.4
l-h-s	2280929	2288328.2
l-h-i	1586502	1589414.8
h-l-c	2468081	2496781.6
h-l-s	1573431	1580786.8
h-l-i	1204845	1206638.4
h-h-c	114841390	118413991.8
h-h-s	64502140	67964923.8
h-h-i	45446258	45954812.2

The best schedules obtained by DE with upgraded initial population were superior in terms of makespan in all cases. For all ETC matrices, except of h-h-s, also the average DE makespan outperformed makespans obtained by scheduling heuristics.

The flowtime obtained by the DE with the upgraded initial population was not the best in all cases. However, the differential evolution managed to optimize makespan and flowtime at once whereas the heuristic algorithms were not able to do that. Also, the value of *lambda* used during the experiment prioritized makespan.

The makespan of optimized schedules is illustrated in Figure 2 and the flowtime of optimized schedules is shown in Figure 3. The series denoted ”DE rand” illustrates the results obtained by DE with random initial population. The series labeled ”DE heur” illustrates in the figures the values of makespan and flowtime obtained



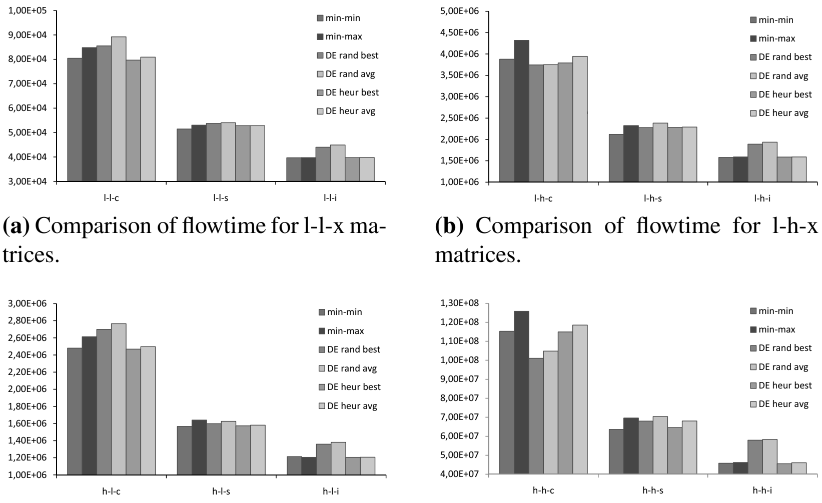
(a) Comparison of makespan for l-l-x matrices.

(b) Comparison of makespan for l-h-x matrices.

(c) Comparison of makespan for h-l-x matrices.

(d) Comparison of makespan for h-h-x matrices.

Fig. 2 Comparison of obtained makespan



(a) Comparison of flowtime for l-l-x matrices.

(b) Comparison of flowtime for l-h-x matrices.

(c) Comparison of flowtime for h-l-x matrices.

(d) Comparison of flowtime for h-h-x matrices.

Fig. 3 Comparison of obtained flowtime

by DE with upgraded initial population. The results are compared to the schedules obtained by min-min and min-max heuristics.

5.3 Comparison with GA

The comparison of the schedules obtained by the DE with schedules obtained by the GA as presented in [10] is shown in the Table 6. We note that the comparison is rather illustrative and indirect since the DE implementetion was not adjusted to perform similar amount of work as the GA from [10] and the weight λ used in the fitness function was also different. Also, the heuristic algorithm with the best makespan is included in the comparison.

Table 6 The comparison of makespan and flowtime obtained by meta-heuristic algorithms DE and GA

(a) Makespan obtained by DE and GA.

ETC	min-max	DE_{avg}	GA_{hier}	GA_{simult}
l-l-c	5310	5271	5310.615	**5226.972**
l-l-s	3327	**3326.8**	3650.316	3515.526
l-l-i	2523	**2502.2**	2676.207	2613.11
l-h-c	273467	270912.4	252907.58	**248466.775**
l-h-s	146953	146759.4	137785.504	**127641.889**
l-h-i	102543	**101254.6**	108047.402	106069.101
h-l-c	164134	161262.2	157255.844	**155251.196**
h-l-s	103321	102440.2	101817.924	**98334.64**
h-l-i	77873	76297.4	77455.085	**75924.023**
h-h-c	7878374	7856042.8	7730973.882	**7610176.437**
h-h-s	4368071	4372414.6	4745284.938	**4359312.628**
h-h-i	2989993	**2953782.6**	3141395.822	3077705.818

(b) Flowtime obtained by DE and GA.

ETC	min-max	DE_{avg}	GA_{rib}	GA_{ss}
l-l-c	84717	**80785.4**	942076.61	920475.17
l-l-s	52935	**52754.8**	616542.78	605375.38
l-l-i	**39679**	39724.6	453399.32	446695.83
l-h-c	4357089	**3983780.4**	35677170.8	34767197.1
l-h-s	2323396	**2288328.2**	15992229.8	15644101.3
l-h-i	1589574	**1589414.8**	13108727.9	13444708.3
h-l-c	2613333	**2496781.6**	28314677.9	27687019.4
h-l-s	1640408	**1580786.8**	17007775.2	16598635.5
h-l-i	**1205625**	1206638.4	12902561.3	12775104.7
h-h-c	125659590	**118413991.8**	1073774996	1048333229
h-h-s	69472441	**67964923.8**	541570911	526866515
h-h-i	46118709	**45954812.2**	376800875	378010732

In the Table 6a, the column GA_{hier} shows the makespan obtained by the GA with the hierarchical fitness function while the column GA_{simult} shows the makespan obtained by the GA with the fitness function with simultaneous optimization of flowtime and makespan.

In the Table 6b, both GA variants used the fitness function with simultaneous optimization of both objectives that perfromed better for the makespan optimization. Two generation replacement startegies were considered in [10] for the GA. The GA with the replace only if better (GA_{rib}) strategy forms the new generation from the best individuals from the union of the set of parent and offspring chromosomes.

The GA with the steady state generation (GA_{ss}) replacement strategy replaces only the weakest individuals in the parent population by offspring chromosomes if they have better fitness.

We can see that the GA_{simult} evolved the schedule with the best makespan 8 times while the DE obtained the schedule with the best makespan 4 times. However, the DE obtained flowtime better than the GA in all cases and it obtained the best flowtime for 10 out of 12 ETC matrices.

6 Conclusions

This paper presents an algorithm for scheduling independent tasks on heterogeneous distributed environments based on the differential evolution. The algorithm was implemented and the experimental results suggest that it can deliver competitive results. With the random initial population, the algorithm managed to optimize schedules for few ETC matrices so that the flowtime was best.

Much better results were obtained when we upgraded the initial population with candidate solutions obtained by the heuristic algorithms. In such case, the algorithm managed to exploit the different sub-optimal solutions provided at the beginning and converged to better schedules.

In comparison to a recent genetic algorithm for the job scheduling problem, the DE has found schedule with better makespan several times but it was outperformed by the GA for some other ETC matrices. However, the schedules obtained by the GA featured better flowtime.

Presented algorithm has a number of parameters including C, F and λ. Fine tuning of DE parameters is subject of our future work.

Acknowledgements. This work was supported by the Czech Science Foundation under the grant no. 102/09/1494.

References

1. Ali, S., Braun, T., Siegel, H., Maciejewski, A.: Heterogeneous computing (2002)
2. Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., Freund, R.F.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems (2001)

3. Tracy, M.M., Braun, T.D., Siegel, H.J.: High-performance mixed-machine heterogeneous computing. In: 6th Euromicro Workshop on Parallel and Distributed Processing, pp. 3–9 (1998)
4. Fernandez-Baca, D.: Allocating modules to processors in a distributed system. IEEE Trans. Softw. Eng. 15(11), 1427–1436 (1989)
5. Munir, E.U., Li, J.-Z., Shi, S.-F., Rasool, Q.: Performance analysis of task scheduling heuristics in grid. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3093–3098 (August 2007)
6. Izakian, H., Abraham, A., Snasel, V.: Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments. In: International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, pp. 8–12 (April 2009)
7. Ritchie, G., Levine, J.: A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments. In: Proceedings of the 23rd Workshop of the UK Planning and Scheduling Special Interest Group (December 2004)
8. YarKhan, A., Dongarra, J.: Experiments with scheduling using simulated annealing in a grid environment. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, pp. 232–242. Springer, Heidelberg (2002)
9. Page, A.J., Naughton, T.J.: Framework for task scheduling in heterogeneous distributed computing using genetic algorithms. Artificial Intelligence Review 24, 137–146 (2004)
10. Carretero, J., Xhafa, F., Abraham, A.: Genetic algorithm based schedulers for grid computing systems. International Journal of Innovative Computing, Information and Control 3(7) (2007)
11. Abraham, A., Liu, H., Grosan, C., Xhafa, F.: Nature Inspired Meta-heuristics for Grid Scheduling: Single and Multi-objective Optimization Approaches. Studies in Computational Intelligence, vol. 146, pp. 247–272. Springer, Heidelberg (2008)
12. Munir, E.U., Li, J., Shi, S., Zou, Z., Rasool, Q.: A performance study of task scheduling heuristics in hc environment. In: An, L.T.H., Bouvry, P., Tao, P.D. (eds.) MCO. Communications in Computer and Information Science, vol. 14, pp. 214–223. Springer, Heidelberg (2008)
13. Freund, R.F., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J.D., Mirabile, F., Moore, L., Rust, B., Siegel, H.J.: Scheduling resources in multi-user, heterogeneous, computing environments with smartnet. In: Heterogeneous Computing Workshop, vol. 0, p. 3 (1998)
14. Shoukat, M.M., Maheswaran, M., Ali, S., Siegel, H.J., Hensgen, D., Freund, R.F.: Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of Parallel and Distributed Computing 59, 107–131 (1999)
15. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)
16. Jongen, H.T., Meer, K., Triesch, E.: Optimization Theory. Kluwer Academic Publishers, Dordrecht (2004)

A Lightweight Approach to Distributed Network Diagnosis under Uncertainty

Javier García-Algarra, Pablo Arozarena, Sergio García-Gómez,
Alvaro Carrera-Barroso, and Raquel Toribio-Sardón

Abstract. Management applications have not kept the fast changing pace of networks and services and still rely on centralized and deterministic approaches. Besides, distribution and uncertainty are intrinsic issues in the telecommunications environment. Therefore, new approaches to network and service management have to be explored that can deal with these challenges.

In this paper a lightweight collaborative framework for network troubleshooting is presented. This framework is based on multi-agent platforms and probabilistic techniques and it has been prototyped and applied to three different network environments. A summary of the most relevant results obtained and conclusions reached is also provided.

Keywords: Bayesian Network, Multi-Agent System, Network Troubleshooting, Uncertainty, Collaborative diagnosis.

1 Introduction

Telecommunication networks are growing in size and complexity, with a rich blend of services being deployed on top of them. All kind of organizations, from small companies, to NGOs, academic institutions or *Fortune 500* corporations use

Javier García-Algarra · Pablo Arozarena ·
Alvaro Carrera-Barroso · Raquel Toribio-Sardón
Telefónica Investigación y Desarrollo,
Emilio Vargas 6, 28043 Madrid, Spain
e-mail: {algarra,pabloa,alvaroc,pabloa,raquelt}@tid.es

Sergio García-Gómez
Telefónica Investigación y Desarrollo,
Parque Tecnológico de Boecillo,
47151 Boecillo (Valladolid), Spain
e-mail: sergg@tid.es

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 95–116.
springerlink.com

Internet as an infrastructure that supports their operations. However, the power and flexibility of this kind of solutions has some drawbacks. For instance, since there are multiple players and no central authority, uncertainty becomes a burden.

Network Management is facing a change of paradigm after a long period of stability. For decades, well established reference models like ITU-T TMN [1] or more recently TMF NGOSS [2], have been the guide for the design of commercial and in house systems, with a wide range of applications for the telecom industry and corporate networks. IETF SNMP [3] has played a similar role for small and medium size business solutions.

Classical architectures like TMN have a common underlying design principle: the state of all existing entities can be fully known at any given moment in time. A hierarchy of layers allows a well engineered distribution of functions. The five big management areas (FCAPS: Fault, Configuration, Accounting, Performance, Security) guarantee that the network is always under control.

This deterministic approach has proved very useful when the entire infrastructure belongs to the same domain, as it is the case within a telecom operator network, when well defined interfaces allow the interconnection of different domains or when all players share a common “management language”.

Another important feature of the classical model is its centralized design. Huge network inventories, extremely complex end to end monitoring applications or even trouble ticketing workflows behave as part of a “Big Brother” that needs every piece of information to react when something unexpected happens.

We can compare this situation to the state of development of classical mechanics by the end of XVIII Century, when it seemed that the Universe was like a perfect clock. Pierre-Simon Laplace wrote in 1814, in the *Essai philosophique sur les probabilités* :

“Une intelligence qui, à un instant donné, connaîtrait toutes les forces dont la nature est animée et la situation respective des êtres qui la compose embrasserait dans la même formule les mouvements des plus grands corps de l'univers et ceux du plus léger atome; rien ne serait incertain pour elle, et l'avenir, comme le passé, serait présent à ses yeux”. (Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective positions of the beings which compose it, if moreover this intelligence were vast enough to submit these data to analysis, it would embrace in the same formula both the movements of the largest bodies in the universe and those of the lightest atom; to it nothing would be uncertain, and the future as the past would be present to its eyes).

When the number of state variables reaches a critical threshold, the status of a system it is no longer computable due to scalability problems. Telecom networks share this property, since the amount of managed entities grows so quickly that is impossible for centralized management solutions to keep the pace. Laplace stated that, as such mighty intelligence does not exist, probability theory had to be developed to deal with this sort of problems.

There is another important source of uncertainty due to the complex nature of telecom networks [4]-[7] that make emergent behavior arise as a result of the interaction of their technical and social building blocks. Moreover, unpredictable socially driven events can have dramatic effects. Therefore, uncertainty cannot be underestimated or dismissed as an undesirable collateral effect. It must be indeed considered as an intrinsic property of telecom systems and it should be taken into account to avoid expensive and slow workarounds.

In particular, troubleshooting is one of the Network Management areas more sensitive to uncertainty [8]. Based on remote access to testing and information capabilities, Network Management tries to come up with a conclusion but, if the information is incomplete or inaccurate, the process may get blocked. Usually, trying to diagnose affected services is very complicated, involving dozens of elements of heterogeneous technologies, and eventually requiring human intervention to solve the puzzle. As human expertise is a scarce and expensive resource, the research community is endeavoring to build systems that emulate network engineers working under uncertainty.

Bayesian Networks (BN), a term coined by Judea Pearl [9], are based on probability theory. The problem domain is represented as a directed acyclic graph where the nodes represent variables, and the arcs, conditional dependencies between them. For example, in the field of network diagnosis, a BN could represent the probabilistic relationships between the root failures and the symptoms observed from the network, such as alarms, the result of executing tests, etc. Graphs are easy to work with, so BNs can be used to produce models that are simple for humans to understand, as well as effective algorithms for inference and learning [10]. BNs have been successfully applied to numerous areas, including medicine, decision support systems, and text analysis [11].

We present in this paper a lightweight, collaborative and distributed approach for network troubleshooting that relies on Bayesian inference to diagnose problems. Our approach is based on three principles and three design decisions:

- The first principle is that any new solution must be neutral and fit on any OSS (Operations Support Systems) map. One common mistake in the management field is to expect that the surrounding IT systems have to adapt to our needs, and this assumption is a source of delays and expenses. The second one is that deployment disruption must be minimal, so it is not necessary to replace any previous system. The third one is the fact that uncertainty is unavoidable, so new systems have to be able to properly deal with it.
- The first design decision is that systems have to be distributed, and grow organically as a part of the network and not as an external "watchman". The second one is that semantics is part of the human knowledge and so systems must be based on semantics just from the beginning. The third one is the application of probabilistic techniques to deal with uncertainty. The combination of BNs with semantic technologies, which is an active research field [12]-[13], provides a high degree of flexibility to diagnosis systems.

In addition to these principles and design decisions, there is also an economic constraint: systems must be cheap to develop, deploy and maintain. Economy is a key issue since network operation costs have not dropped at the same rate than equipment prices and represent a major problem for service providers [14]. All these considerations have been applied to three different scenarios:

- KOWGAR, an online troubleshooter to help the final users of a geographically distributed corporate network.
- KOWLAN [15], an automatic diagnosis system for the Ethernet/VPN commercial service of Telefónica España (MACROLAN).
- MAGNETO [16], the application of these concepts to Home Area Networks, where two different management domains (ISP and Home Area Network) have to cooperate to reach valid conclusions.

The paper is organized as follows: after this introduction section 2 describes relevant related work. Section 3 provides a high level overview of the system architecture, describing its main components. Section 4 goes on to describe how diagnosis information is modeled, with particular emphasis on BN modeling and self learning features. Then in section 5 the most relevant aspects of the three different scenarios addressed are summarized. A brief description of our main implementation decisions is given in section 6 and, finally, section 7 provides some conclusions and suggests possible enhancements.

2 Related Work

There are many studies that address the scalability and reliability problems of traditional centralized network management. Such is the case of a number of recent research projects in autonomic networking, which have explored architectures for embedding management functionality in network elements in order to decentralize resource and network management. Typical projects that have taken this approach are ANA [17], 4WARD [18] and EFIPSANS [19]. While these projects introduce various architectural control loop abstractions in order to define generic management architectures that could be applied in a wide range of network scenarios, we follow a lighter approach which focuses on network troubleshooting for also a wide range of network scenarios. However, our solution can be easily integrated as part of an autonomic management control loop.

On the other hand, there are several studies in the area of distributed automatic network troubleshooting. For instance, AutoMON [20] uses a P2P-based solution to test network performance and reliability. In this solution, distributed testing and monitoring nodes are coordinated by using a DHT (Distributed Hash Table) algorithm, which helps in locating resources or agents. Here the nodes do not cooperate nor use historical information about failures. In Connected Home [21] fault management is performed in a distributed fashion, following an agent-based approach, but only focusing on the Connected Home scenario. DYSWIS [22] presents an “Architecture for Automated Diagnosis of Networks” that consists of error detection and diagnosis nodes. The first ones look for network failures by passive traffic monitoring and active probing, while the second ones determine the failure root cause by using historical information and orchestrating the execution of active tests. Network dependency relationships are encoded as rules. In [23] “A Multi-agent System for Cooperative Network-Fault Management” is described, having specialized agents to perform fault detection, diagnosis and operations to recover from network faults. MADEIRA [24] introduces a distributed architecture for dynamic devices management. The use of a dynamic hierarchy allows the system to adapt to network failures and state changes, something that is considered essential for future large scale networks which must exhibit adaptive, decentralized control behavior. The main drawback of these approaches is that they are not designed to deal with uncertainty.

To tackle the problem of uncertainty, several authors have explored the application of probabilistic techniques to network management [25-29]. For fault diagnosis, many of these studies use Bayesian Inference [30-32] to conclude the cause of the observed network problems. In fact, research by Nokia Networks and University of Malaga [33] compared three reasoning algorithms for the automation of mobile network troubleshooting and concluded that BNs was far better suited for the use cases identified. On the other hand, CAPRI [34] defines a Common Architecture for Distributed Probabilistic Internet Fault Diagnosis. The whole picture is similar to the one in DYSWIS, but it uses BNs instead of rules to infer the root cause of the failure. Although CAPRI has inspired much of our work, our proposed architecture is more flexible and open to address any network diagnosis scenario, not just Internet faults.

3 Architecture Overview

This paper presents a generic architecture for automated distributed network fault diagnosis. Based on the Multi-Agent System (MAS) paradigm [35], this architecture can be easily adjusted to deal with different new network and service scenarios. The key points lie on the combination of using Bayesian inference and semantic knowledge in the diagnosis process.

A MAS is composed of largely autonomous and decentralized components, called agents, that cooperate together in order to solve a common goal. In the MAS presented in this paper for automated distributed network fault diagnosis, there are several types of agents defined, each type with a specific roles or capacities, but all of them sharing the same objective: to come up with a diagnostic of high confidence. The main types of agents are the following:

- *Fault detection agents:* They are in charge of detecting network faults and triggering the automatic diagnosis procedure.
- *Diagnosis agents:* They orchestrate the diagnosis process for a specific network scenario. For that purpose they communicate with other agents to gather relevant information.
- *Observation agents:* Their mission is to get evidences from the managed networks and services by performing tests, analyzing alarms, etc. These agents either have interfaces with network resources and their Management Information Bases (MIBs) or make use of external testing tools and services.
- *Belief agents:* They provide beliefs (not evidences or facts) of the status of a given network or service component. These agents have Bayesian knowledge embedded in their knowledge repository, and perform Bayesian inference to conclude a belief of a certain node based on the evidences they may get from

other observation agents. In other words, these agents help the system to split the BN knowledge between different agents.

- *Knowledge agent:* This agent is in charge of the diagnosis knowledge distribution to all interested agents. It also improves the diagnosis results by processing the past validated diagnosis reports to adjust the values of the Bayesian Conditional Probability Tables (CPTs).

Figure 1 shows how the main agents of this multi-agent system relate to each other.

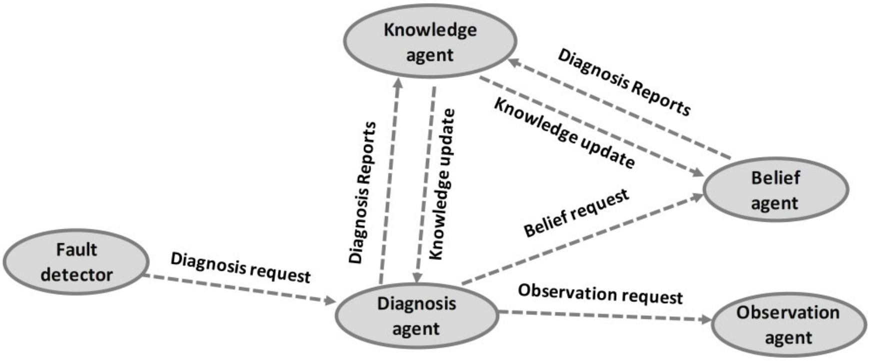


Fig. 1 Communications between agents

This architecture can be customized depending on the specific needs of each scenario. Thus, for each particular scenario different BNs are defined and specialized agents are developed.

As it has already been stated, the diagnosis agents orchestrate the diagnosis process for a specific network scenario, modeled by means of a BN ontology. Belief and observation agents publish the evidences and beliefs they can provide together with the cost associated to them. On the other hand, diagnosis agents discover them using a publish-subscribe mechanism.

Upon receiving a diagnosis request, containing the detected error, a diagnosis agent starts a diagnosis process. First, it creates an instance of the BN for the current scenario. Then, it gathers all known related evidences for the diagnosis (detected error plus other evidences used in previous diagnosis procedures that are still valid according to a time threshold). Finally, it starts an iterative algorithm that encompasses the following actions:

1. Inserting into the BN all known evidences.
2. Performing bayesian inference.
3. Evaluating the diagnosis result: If a hypothesis for the cause of the error reaches a certain confidence threshold, the diagnosis stops. Otherwise, go to 4.
4. Selecting the best action to perform (either belief or an observation request), taking into account its cost and if there is enough data to request it. If no action is found (due to the fact that all possible actions have already been executed),

the diagnosis stops. Otherwise, request the selected action to the appropriate agent. When it gets the results, go to 1.

Note this algorithm continues until either the expected confidence is reached or there are no additional tests available. The selection of tests takes into account the cost of each action and additional tests can be smoothly added by creating new observation agents.

4 Knowledge Modeling

The agents described in the architecture use BNs to infer, given a set of observations, the most likely hypothesis of failure in the service or network. The definition of the BNs' structure and CPTs is carried out by means of a tight collaboration process between the platform experts and the technicians that have a solid knowledge and experience in the domain to be modeled. In a first step, based on the actual diagnosis procedure, the causal dependencies are outlined. Then, a refinement process leads to a BN ready to be used in the diagnosis process. The platform also features self-learning capabilities, as described below.

It is important to stress that there exists one BN per network topology or scenario. Besides, some extra information that is very useful during the diagnosis process (confidence thresholds, parameters required to execute tests, timeouts, etc.) has to be modeled.

Both the BNs and the additional information are modeled by means of an ontology. For this purpose, some classes as Hypothesis, Observer, Threshold or Probability are defined and their instances are particular elements of the described BN (Figure 2). On the other hand, the ontology also provides the language that the agents use to communicate, together with relevant information about the diagnosis, as the observation results, or the beliefs available at a given point in time.

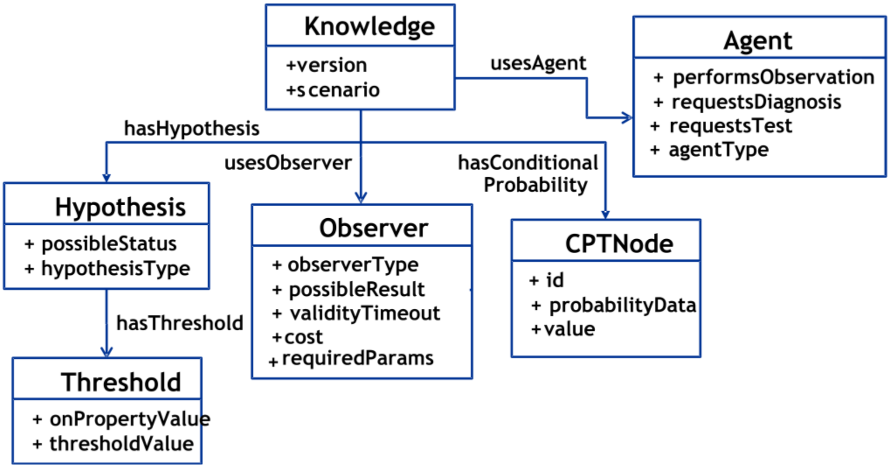


Fig. 2 Bayesian Knowledge modeling

The agents that need a particular BN load its instance on start-up and subscribe to be notified when it changes. In this way, it is possible to modify the behavior of the diagnosis agents while they are running.

4.1 Bayesian Network Partitioning

In a complex scenario, such as network troubleshooting, it may be necessary to distribute diagnosis intelligence across the physical network, splitting the inference process into several smaller processes that make it more scalable and efficient. To achieve this goal, instead of using a centralized BN, a smarter approach is to partition the whole domain in smaller BNs or clusters [36].

Following this principle, different elements in different parts of the network may have different views and knowledge. For instance, some agents may be specialized in diagnosing network problems while others in service problems, exchanging their conclusions to cooperatively reach a valid diagnosis. In other words, the single BN that exists in a centralized solution would be fragmented and distributed to an overlay agent network.

A variation of the Virtual Evidence Method (VEM) [37] is used for this purpose. Both diagnosis and belief agents make use of that algorithm to share a belief on a certain node. With this method, each agent only needs to be aware of and reason on a fragment of the original BN, while the complete diagnosis is done by cooperative inference among diagnosis and belief agents. This way the diagnosis process becomes more scalable, since computing resources are distributed across the physical network. It also facilitates reusing knowledge in several diagnosis processes that share common parts of the BN. For example, different service diagnosis processes may share a common network topology and also the corresponding BN that models network problems diagnosis. Moreover, if the BN is partitioned according to the physical network topology, it also enables mapping the diagnosis process to the different network domains.

4.2 Bayesian Network Self Learning

When modeling a BN two things have to be defined:

- The structure of the BN, i.e. the nodes or variables and the arcs or casual dependencies between them.
- The parameter values of the Conditional Probability Tables (CPT), i.e. the values of the dependencies defined in the BN structure.

When there is a big amount of historical data, the structure of the BN can be created by using structural learning algorithms such as K2 [38] and tree augmented Naïve Bayes [39]. The main drawback of these algorithms is that, since they are just based on probabilities, they may come up with wrong casual dependencies. That is the reason why, when there is expert knowledge available, the structure of the BN is usually defined by experts in the domain. This is the approach followed in our work, where the initial BN modeling for each scenario has been performed manually with the help of the service and network experts.

However, one significant goal in our work is to allow the diagnostic intelligence to be able to self adapt and improve over time. To achieve this, the system must be able to learn from past actions, something which requires a feedback loop to validate past diagnoses. A possible solution is to request human feedback from network operators but this is not very efficient. Therefore, the preferred solution is automatic validation of diagnosis results. This is a very challenging task, since there is not a straightforward way to assess diagnosis results. Based on this feedback, either manual or automatic, parametric learning algorithms are used to improve diagnosis quality. By means of these parametric learning algorithms, the link weights in the CPTs are updated, but not the BN structure.

In particular, the Expectation Maximization algorithm (EM) is used, which is an efficient algorithm that finds maximum likelihood estimates of parameters in probabilistic models [40]. EM follows an iterative approach that firstly estimates the missing values in the input data representing previous diagnoses and then recalculates, using statistical methods, the BN's CPTs. The initial estimation of missing values is very relevant in network troubleshooting since for some diagnoses there may only be a subset of the possible evidences available.

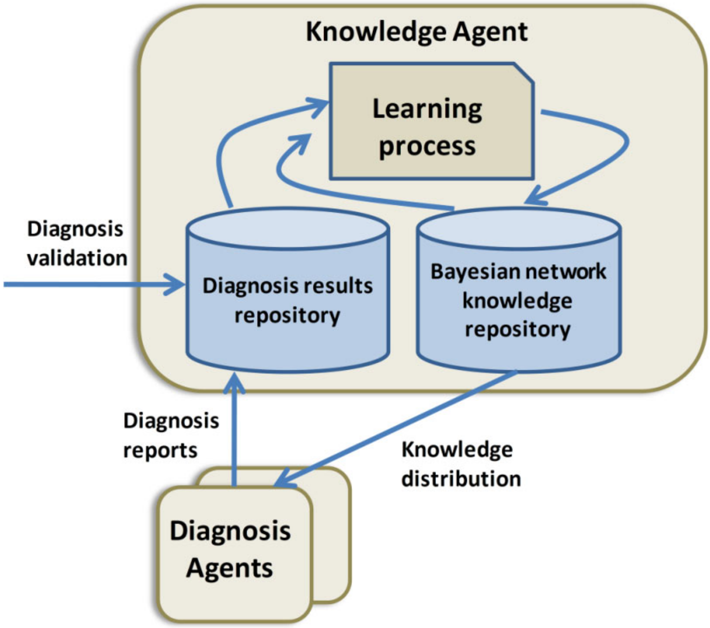


Fig. 3 Self-learning loop

In summary, self-learning comprises the following steps (Figure 3):

1. Initially, the BN is manually created based on diagnosis knowledge provided by experts on the domain (basically skilled network operators).
2. The BN is then distributed to all diagnosis agents.
3. Whenever a diagnosis process is completed, diagnosis results are reported to a specialized agent that stores them in a knowledge repository.
4. Diagnosis reports are then validated to indicate whether the result was right or wrong. This validation can be either manual or automatic.
5. CPT values are periodically updated by executing a self-learning process that accesses the validated diagnosis reports and the BN knowledge available in the common repository.
6. Once the BN is updated with the new CPT values, the process continues with step 2.

Note that, although self-learning could be executed in a distributed way, we believe that it is more convenient to conduct this process following a centralized approach due to the following reasons:

- The self-learning algorithm can use a much larger data set, thus producing more accurate results.
- A powerful server and specialized resource-consuming tools may be needed to execute the self-learning process. However, these resources may not be available in all domains.

5 Application Scenarios

5.1 *Bayesian Diagnosis in Corporate Networks*

KOWGAR is a proof of concept to test the suitability of a distributed Bayesian diagnosis system, applied to our company's corporate Intranet. The system targets a reduced set of problems related with web navigation across internal and external web sites. The scenario is very common in any kind of corporate Intranet with geographically distant locations, communicated among them through a Virtual Private Network.

KOWGAR was designed to help end users with a quick and cheap deployment. A diagnosis process is automatically triggered by a Firefox plug-in when the browser detects a problem. Then, the user only has to check the final diagnosis result.

Two different types of possible HTTP connections can be distinguished:

- Connections to HTTP servers within the intranet.
- Connections to HTTP servers located outside the TID's intranet, i.e. in the Internet.

The universe of root causes is selected prior to the construction of the BN. Some of them are as follows:

- Local misconfiguration,
- Link failure
- Routing failure
- DNS server unreachable
- DNS information incorrect,
- Destination host unreachable
- Destination port unreachable
- Destination application unavailable.

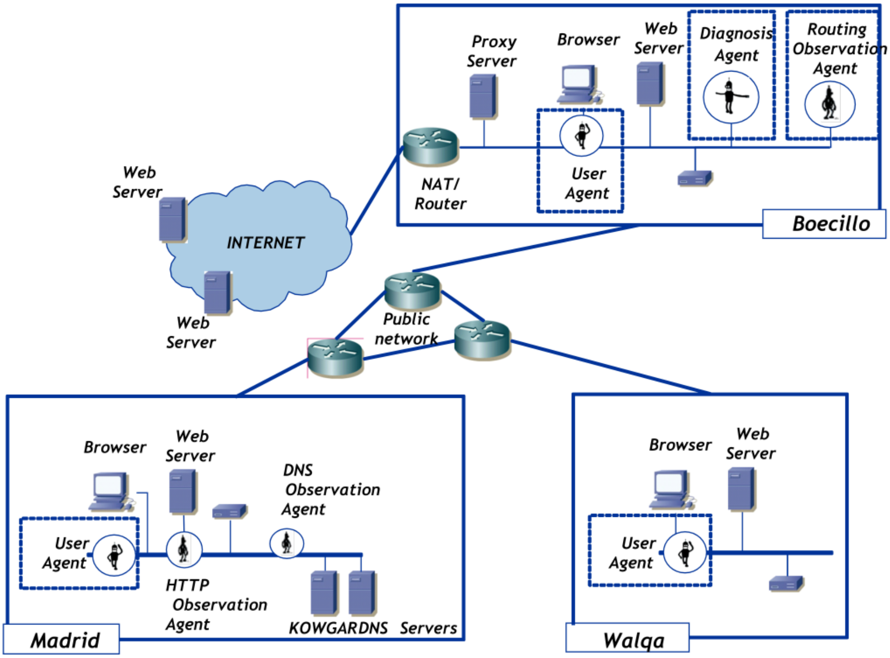


Fig. 4 KOWGAR testbed scenario and agents deployment

For KOWGAR, a new type of agent has been defined: the User Agent. This type of agent has embedded the roles of fault detection, diagnosis and observation agent in the sense that it can detect an http service failure and trigger automatically a diagnosis procedure, it can perform a set of local tests (such as a local connectivity test) and it can perform bayesian inference to diagnose local problems. Finally, if it cannot reach a valid conclusion, it can delegate the diagnosis procedure to a regional diagnosis agent that gathers all observations from all the user agents within its region. This User Agent performs a set of basic testing procedures, and if everything is working fine, sends a request to the Diagnosis Agent, the agent in charge of orchestrating KOWGAR's actions. This agent knows the BN and the associated ontology that explains which kind of test must be executed to gather the different possible observations. So, it invokes three different Observation agents: routing, DNS and HTTP. In this case, the BN concludes that the failure has been likely caused by server congestion. The information can be displayed from the browser just by clicking the plug-in icon.

There were three locations involved in the experiment: Madrid, Valladolid and Walqa (Huesca). To simulate DNS failures without disrupting the daily Company operations, two mock DNS servers were installed. Figure 4 shows the testbed and an example of what happens when a final user in Valladolid subnetwork is trying to access a web server in Madrid that is refusing connections due to high load. When an error is detected Firefox triggers the plug-in, which in turn activates the User Agent.

This simple example shows one of the advantages of distributed diagnosis: problems are detected locally if possible. If the User Agent detects a network wire failure, for instance, no further tests would be necessary and the process stops. For example, it is interesting to note the behavior of the Diagnosis Agent in a different scenario, when there is a problem in the routing tables of a Valladolid router. In this case, the evidences gathered by the User and Routing Agents would be enough to reach this conclusion without performing additional tests.

Since KOWGAR is just a short term experience developed in three months, no self-learning algorithms were implemented, but the distribution of manual changes in the BN and the ontology was successfully tested.

5.2 *Fault Diagnosis in Virtual Private Networks*

MACROLAN is Telefónica España's solution to build Virtual Private Networks (VPN) connecting multiple enterprise sites over Ethernet based access networks. MACROLAN supports service speeds from 2 Mbit/s to 1 Gbit/s. By using standard L2 and L3 VPN technologies, MACROLAN enables geographically distant customer sites to communicate as if they belong to the same LAN, in terms of speed, reliability and transparency.

The MACROLAN service is built on top of a diverse set of network technologies. The local loop from the customer location to the Central Office (CO) can be either fiber with media converters, when available, or copper, in rural areas. Synchronous Digital Hierarchy (SDH) circuits allow to extend the distance of this access segment when the closest MAN access point is in a different CO. MACROLAN traffic is aggregated into province-wide Metropolitan Ethernet networks (MANs) and then into an IP backbone network that provides national coverage.

MACROLAN poses an interesting challenge since it involves a number of different technologies. Diagnosing service errors requires a high degree of skills, experience and ability since it involves accessing information from half a dozen different OSSs simultaneously. The goal is then to capture the expertise from human operators, model it in BNs and thus reduce manual intervention for the most common types of failures. There are two additional: the system should be deployed without any change in the existing OSSs and it should be running on a live network after a month of development.

KOWLAN is the solution designed to help MACROLAN technicians to diagnose access problems. It provides the capacity to diagnose automatically reported customer complaints. When a customer reports a problem in the service, a trouble ticket (TT) is generated. The only pieces of information available about the failure are the circuit identification and the reported symptom (packet loss, slow communication, etc). This TT is automatically fed into KOWLAN, and a diagnosis procedure is triggered without human intervention. Also, depending on the most probable cause of failure, the ticket can be transferred to the appropriate company repair unit. Besides the automatic network diagnosis, KOWLAN also allows human operators to trigger a manual diagnosis and to access historical reports.

Six different service network scenarios were identified, one for each type of access technology depending on the combination of fiber, copper and the existence of SDH path. In order to automatically diagnose each one of them it was necessary to:

- Model a BN for each scenario with the aid of expert knowledge. Knowledge capture required only two working days of a team of expert technicians and two engineers.
- Create inventory agents to get, for a particular circuit identifier, its full circuit description (topology, scenario, equipment data and configuration). In order to achieve this, these agents interface with corporate inventories.
- Create an agent that identifies the scenario for a particular TT, so that the diagnosis request is sent to the appropriate diagnosis agent.
- Create a fault detection agent that polls periodically the TT system so, when a new TT is assigned to the MACROLAN technical center, a KOWLAN diagnosis procedure is triggered without human intervention
- Create specific observation agents that perform tests or analyze OSS alarms. In the implementation of these agents, scripts and tools previously created by the MACROLAN operators have been reused.
- Create a user interface agent to enable the OSS operator to request diagnosis on a circuit, display its details and evaluate the accuracy of the diagnosis. This evaluation may be useful for further refinement of the BN parameters by using self-learning mechanisms.
- Create a persistent agent to store diagnosis reports.

KOWLAN architecture can be seen in Figure 5. For simplicity, only observation agents for one of the six service network scenarios are shown.

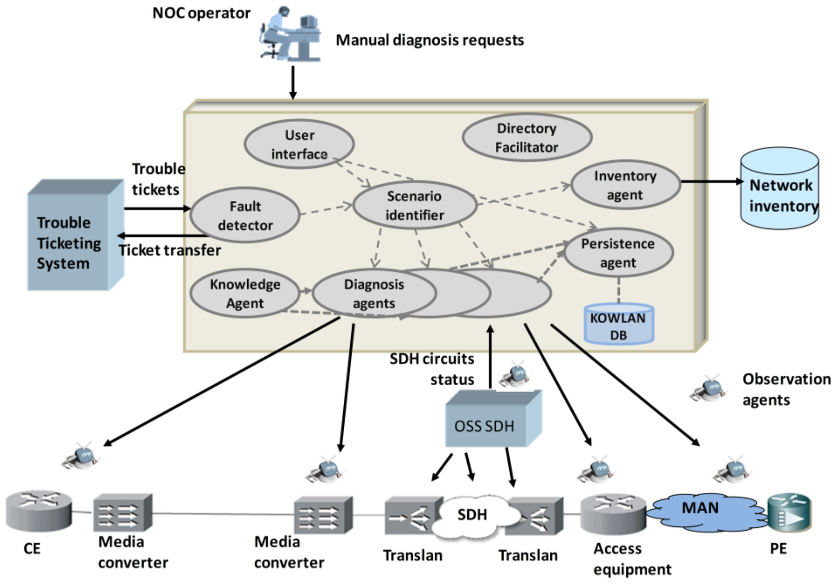


Fig. 5 KOWLAN architecture for a specific network scenario

The adaptation of our generic automatic network troubleshooting architecture to KOWLAN took only four months, thanks to the adaptability of its architecture.

KOWLAN handles approximately 100 TTs per day, and another 300 manual diagnoses triggered by network administrators. Each diagnosis report takes on average 2 minutes 12 seconds, and in 98% of the cases this time is inferior to 4 min 30 seconds. The time needed to get a diagnoses report depends on the scenario complexity and the number of tests to be executed. Most of this time is consumed executing tests rather than in the inference process. This means a significant improvement in relation with the manual diagnosis. Since its installation, an average reduction of 23 minutes has been observed in the diagnosis time. This has on its turn contributed on the reduction of the service recovery time in 54 minutes. Thanks to this, human intervention has been reduced in a 30 %.

TTs with reliable diagnosis reports can be enabled to be dispatched automatically to the corresponding repair unit. Although this functionality is not currently enabled, it is estimated that its activation will introduce a further service recovery time reduction of approximately 32 minutes.

Even though KOWLAN does not execute all possible tests nor it works with complete and reliable information about the state of the related service elements, in near 40% of the reported TTs it concluded a diagnostic of high reliability, higher than 95%. Table 1 shows the classification of KOWLAN diagnosis reports depending on their reliability. As new system observations are introduced, it is expected that the percentage of reliable diagnosis reports will be higher than 60%.

Table 1 Classification of KOWLAN diagnosis reports

Conclusion	Reliability	Frequency
Reliable	>95%	38%
Probable	40% - 90%	18%
Uncertain	<40%	44%

On the other hand, KOWLAN presents the capability of self learning to adjust the CPT values of the BN defined for each service network scenario. OSS operators can validate the diagnosis results provided by KOWLAN by means of a user interface. Currently operators are only validating a small percentage of the diagnostics, but once they proceed injecting this information into the system, KOWLAN will achieve even better diagnosis results.

The key for the successful application of Bayesian diagnosis in KOWLAN has been the high involvement of MACROLAN maintenance team in the project, since knowledge capture has shown to be very accurate.

5.3 Management of the Outer Edge

Management of outer edge networks, such as corporate or home area networks, is a very complicated task, not only because of technical constraints, but also due to legal, regulatory, security and commercial issues. In the MAGNETO project, a fault diagnosis prototype is being developed to address Home Area Network (HAN) troubleshooting.

Management of HANs represents a huge challenge for telecom operators since it has to combine the management of different network domains. Some of these domains are under the control of the telecom operator, while the rest belong to the end customer. This means that the level of detail in the information available from each domain may be very different. For example, full knowledge about a HAN configuration and status cannot be taken for granted.

Current management architectures address HAN management from a centralized perspective, where management tasks are performed in a management system that remotely accesses customer equipment using protocols such as TR-069 [41]. In order to allow some degree of autonomy to the management of HANs, MAGNETO is exploring a distributed architecture where management tasks are locally executed improving efficiency and reducing the burden on centralized servers.

Besides, an important MAGNETO feature is its capability of self diagnosing problems affecting HAN users. Note these problems may be caused by failures originated both inside and outside the HAN. Therefore, self diagnosis in MAGNETO may require cooperation between management agents placed in different HANs or even network domains. For example, MAGNETO agents sitting on HAN equipment may reach some conclusions based on the evidences locally available but may also need to cooperate with agents in the ISP network or other HANs to exchange their views on the problem in order to reach a valid diagnosis. MAGNETO fault diagnosis functionality follows a similar approach to the one already described for KOWGAR and KOWLAN, relying on bayesian inference to diagnose the cause of service and network failures, with the diagnosis knowledge provided by expert network operators.

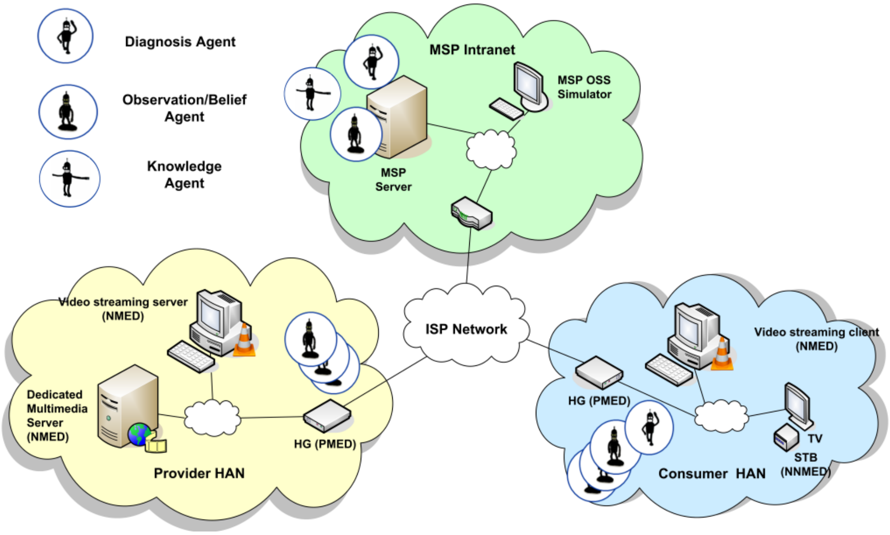


Fig. 6 MAGNETO deployment in a HAN environment

In MAGNETO, some parts of the overall BN can be executed in HAN devices (like home gateways, set top boxes, etc.) while other parts run inside the ISP's network. Each of these management domains will have a different perspective on the problems being diagnosed and will exchange information about their conclusions to cooperatively reach a valid diagnosis. For that purpose, MAGNETO will make heavy use of BN partitioning.

Figure 6 depicts a possible deployment of MAGNETO agents for a particular multimedia content sharing scenario that involves two HANs. As can be seen in the picture, there are several agents deployed at different domains, from the HAN to the ISP environment (note we call MSP to a MAGNETO enabled ISP). It is foreseen to have at least one MAGNETO agent inside the HAN, most likely sitting in a well-equipped device like a residential gateway. Additional agents can be deployed in other devices like set top boxes, although this is not compulsory. Nevertheless, due to resource limitations, instead of deploying agents in all multimedia devices it may be preferable to access them remotely from agents in the residential gateway making use of management interfaces. Therefore a Device Adaptation Layer has been developed in MAGNETO so relevant information from the devices can be obtained.

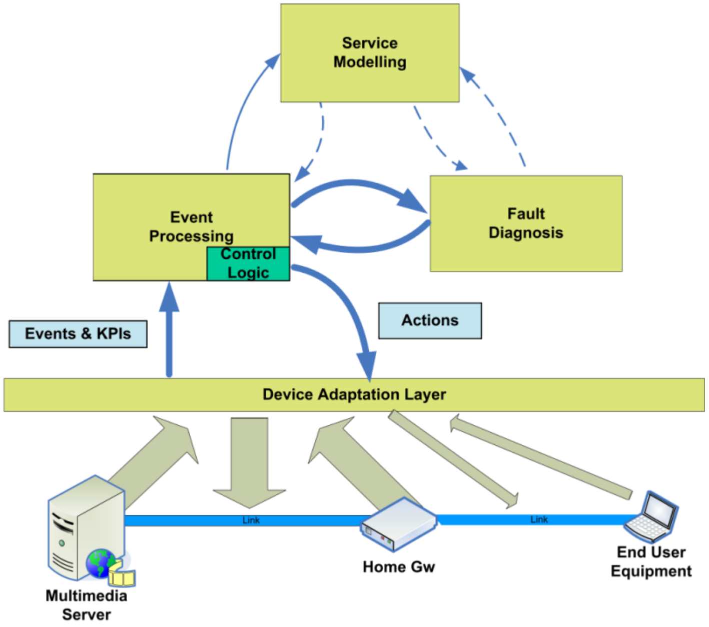


Fig. 7 MAGNETO control loop

In MAGNETO, fault diagnosis is integrated in an autonomic control loop (Figure 7) that allows using diagnosis results to trigger reconfiguration actions intended to solve the problem in hand. It is also foreseen to monitor whether those actions were successful or not to validate diagnosis results and thus feed a self learning loop (Figure 3).

Agents deployed in the MSP network take advantage of existing computing resources like OSS servers or even network equipment. These agents have access to information that is not available inside the HAN and can therefore be used to complete the diagnosis process making also use of the results of the partial inference conducted inside the HAN. In addition, MSP agents can communicate with OSSs belonging to the MSP for two purposes: firstly to request relevant diagnosis information from them and secondly to automatically feedback the results of the diagnosis process, when appropriate, by triggering trouble tickets or alarms.

Another important MAGNETO feature is its capability to improve the diagnosis algorithm by means of a self-learning process. To make this more valuable and accurate, a central server in the ISP network is in charge of gathering historical data about past diagnoses from the different domains involved. Periodically these data will be used to update the BNs applying parametric learning algorithms. The new knowledge acquired will be then distributed to appropriate MAGNETO agents so it can be used for further diagnoses.

6 Implementation Overview

In order to curb development and installation costs, it was decided to only use Open Source tools to develop this framework. An additional requirement was to achieve a portable solution that could be easily deployed in any hardware or software platform.

As a basis for implementing our system, the JADE [42]/WADE [43] multi-agent platform is used. JADE is a complete and sound multi-agent Java platform based on FIPA standards [44] led by Telecom Italia. JADE allows the distributed deployment of Java coded agents with minimal requirements of CPU and memory on servers and devices. Besides, since it is a Java-based platform, portability is guaranteed. Following JADE principles [45], the agents of the presented solution are implemented using behaviors and all the communications between JADE agents are based on FIPA-ACL messages while the communications with external systems are based on XML messages over HTTP or on standard Web Services. WADE environment also provides an important mechanism to manage the deployment of the system agents. To deploy agents inside a HAN, the Apache Felix [46] implementation of the Open Services Gateway Initiative (OSGi) [47] specification is used, since JADE agents can be easily deployed as an OSGi bundle. OSGi offers a simple life cycle management to deploy the required software in the residential gateway inside HAN's.

One drawback of our implementation is the difficulty to deploy agents in very resource limited devices like, for example, set top boxes. In these cases, the preferred approach is to interface with these devices from proxy agents deployed in more resourceful equipment such as a residential gateway.

Although there are more sophisticated ways to represent probabilistic knowledge with ontologies as in [48], the ontology described in previous sections has been implemented using the formalism envisaged by JADE. It is not based on a heavy logical background, but it copes with the system requirements, including an efficient serialization mechanism into FIPA-SL language. Moreover, the first trials implemented with OWL-DL/RDF as exchange language among agents showed up serious efficiency problems for the purposes of this application. The practical implementation was carried out with Ontology Bean Generator [49], a Protégé plug-in for JADE. For each service network scenario, a BN was modeled making use of the GeNIe [50] graphical interface. Then, the GeNIe file obtained was used to populate the BN ontology data needed by the agents to carry out a specific diagnosis procedure.

The Bayesian Inference engine chosen is SamIam [51], a lightweight and fast Java tool developed at UCLA. Since SamIam is based on Java, its integration with JADE is quite easy and allows deploying inference agents in a huge variety of devices. In addition, another Bayesian tool is used in the self learning process: SMILE [52], since its implementation of the EM learning algorithm provides more accurate results. Although it is a quite good implementation, the way it handles confidence has had to be improved, since the weight assigned to previous knowledge is too high compared with input data. For that purpose, a “balanced average” method has been defined that gives equal weight to previously acquired knowledge and the new input data used in the self learning process. This is particularly relevant in the case of conditional probabilities. Therefore, SamIam is used by diagnosis and belief agents to perform on-line fast inference while SMILE is used by the knowledge agent to conduct off-line learning.

The user interface is implemented as a PHP application. Note this interface is mainly meant to be used by network operators. Offering an interface to end users, like HAN customers, would require specialized interfaces that would make user of end user devices capabilities. In order to keep historical logs of diagnosis results and enable self learning, a persistency layer is implemented on top of a MySQL database.

7 Conclusions

Troubleshooting is a very common task in Network Management. This paper has described a lightweight solution for automatic distributed diagnosis based on BNs and built with Open Source components on top of a multi-agent framework. The proposed architecture has been tested on three different scenarios: a corporate Intranet, a telco VPN infrastructure and an experimental testbed for a multi-domain Digital Home proof of concept.

The most relevant results are the flexibility and short time-to-market of the proposed solution. This approach is a good start point for any organization that wants to develop an in house troubleshooting system to cope with the growth and complexity of its own network. Besides, having the business logic encoded in a BN that can be distributed throughout the network at any time is a great advantage, since it allows the addition of new diagnosis capabilities without stopping the system. New diagnosis capabilities can be added very easily by developing specific test agents that would further enhance the accuracy of the diagnosis results.

One significant contribution of our solution to network management scenarios is its capacity to deal with uncertainty, since it is usual to face problems getting information and executing tests from different management systems. The probabilistic approach provides an answer even when not all relevant information is known, although the higher number of evidences are known, the more accurate the diagnosis will be.

Another important benefit is related to scalability, as it has been seen that the multi-agent paradigm offers a good solution to easily deploy decoupled pieces of software in an environment where the overall architecture is not known. Agents can be proactively exploited to locally detect and solve problems, avoiding the participation of more centralized systems in the architecture.

In the future we plan to further explore some of the challenges identified, such as improving self-learning capabilities, smart partitioning of the BNs and automatic feedback on the success of diagnosis.

Acknowledgments. This paper describes work undertaken in the context of the CELTIC MAGNETO project, which is partially funded by the Spanish "Ministerio de Industria, Turismo y Comercio". The authors would also like to thank the MACROLAN Technical Center staff in Barcelona, Spain, for their support during the KOWLAN experience and the Telecom Italia team in charge of the JADE platform for their professional, helpful and kind support.

References

1. ITU-T, Principles for a Telecommunications Management Network, Recommendation M.3010 (1996)
2. Creaner, M., Reilly, J.: NGOSS Distilled – The Essential Guide to Next Generation Telecoms Management. The Lean Corporation (2005)
3. Case, J., Fedor, M., Schoffstall, M., Davin, J.: A Simple Network Management Protocol (SNMP), RFC1157 (1990)
4. Chen, C., Nagi, S., Clack, C.: Complexity and Emergence in Engineering Systems. In: Tolk, A., Jain, L.C. (eds.) Complex Systems in Knowledge based Environments: Theory, Models and Applications, ch. 5, pp. 99–128. Springer, New York (2009)
5. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 251–262. ACM, New York (1999)
6. Spencer, J., Johnson, D., Hastie, A., Sacks, L.: Emergent properties of the BT SDH network. BT Technology Journal 21(2), 28–36 (2003)
7. Cárdenas, S., Mouronte, M.L., Feliú, V., Benito, R.M.: Modeling the topology of SDH networks. International Journal of Modern Physics C 19(12), 1809–1820 (2008)
8. Pras, A., Schönwälder, J., Burgess, M., Festor, O., Pérez, G.M., Stadler, R., Stiller, B.: Key Research Challenges in Network Management. IEEE Communications Magazine 45(10), 104–110 (2007)
9. Pearl, J.: Bayesian networks: A model of self-activated memory for evidential reasoning. UCLA Report CSD-850017 (1985)
10. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall Series in Artificial Intelligence. Prentice-Hall, Englewood Cliffs (2003)
11. Kjaerulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis. Springer, Heidelberg (2008)
12. Laskey, K.B., da Costa, P.C.G.: Uncertainty Representation and Reasoning in Complex Systems. In: Tolk, A., Jain, L.C. (eds.) Complex Systems in Knowledge based Environments: Theory, Models and Applications, ch. 2, pp. 7–40. Springer, New York (2009)
13. Ding, Z.: BayesOWL: A Probabilistic Framework for Uncertainty in Semantic Web. Ph.D. dissertation, University of Mariland, USA (2005)
14. Dogra, R., Orr, S.: Managing Through Challenging Times. Communications Industry Group, http://www.accenture.com/NR/rdonlyres/E40A0832-FB88-45F1-B9CF-F6A14AFA0902/0/ManagingThroughChallengingTimesCommsEALAPOVFinal.pdf

15. García, S., González, J., García, J., Toribio, R., Sedano, A., Buisan, F.: A Multi Agent System for Bayesian Diagnosis in Telecommunication Networks. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 195–198 (2009)
16. MAGNETO: Management of the outer edge, `http://projects.celtic-initiative.org/MAGNETO` (last visited March 2010)
17. Bouabene, G., Jelger, C., Schmid, S.: ANA Blueprint Version 2.0, ANA Project Deliverable D1.4/5/6_v1.1 (2008)
18. Nunzi, G. and Dudkowski, D.: 4WARD Deliverable D-4.2: In-Network Management Concept (2009), `http://www.4ward-project.eu`
19. Chaparadza, R.: Self-Management Workshop. 2nd Concertation Meeting of the FP7 Future Internet Cluster, Brussels (2008)
20. Binzenhöfer, A., Graben, B., Fiedler, M., Arlos, P.: A P2P-based framework for distributed network management. In: New Trends in Network Architectures and Services, Loveno di Menaggio, Como, Italy. LNCS, vol. 3883, pp. 198–210 (2006)
21. Utton, P., Scharf, E.: A fault diagnosis system for the connected home. IEEE Communications Magazine 42(11), 128–134 (2004)
22. Singh, V.: Dyswis: An architecture for automated diagnosis of networks. In: IEEE Network Operations and Management Symposium, NOMS 2008, Salvador de Bahia, Brazil, pp. 851–854 (2008)
23. Garijo, M., Cáncer, A., Sánchez, J.J.: A Multiagent System for Cooperative Network-Fault Management. In: Proceedings of the First International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, PAAM 1996 (1996)
24. Leitner, P., Collins, S., Fahy, C., Zach, M., Leitner, M.: Fault Management based on peer-to-peer paradigms. In: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Management, Munich, Germany (2007)
25. Badonnel, R., State, R., Festor, O.: Probabilistic Management of Ad-Hoc Networks. In: 10th IEEE/IFIP Network Operations and Management Symposium NOMS 2006, Vancouver, Canada, pp. 339–350 (2006)
26. Ding, J., Krämer, B., Xu, S., Chen, H.: Predictive Fault Management in the Dynamic Environment of IP Networks. In: Proceedings IEEE Workshop on IP Operations and Management, pp. 233–239 (2004)
27. Brunner, M., Dudkowski, D., Mingardi, C., Nunzi, G.: Probabilistic Decentralized Network Management. In: Proceedings IEEE INM 2009, Hofstra University, Long Island, New York, USA, pp. 25–32 (2009)
28. Sahin, F.: A Bayesian Network Approach to the Self-organization and Learning in Intelligent Agents. Ph.D. dissertation, Virginia Polytechnic, USA (2000)
29. Ding, J., Jiang, N., Li, X., Krämer, B., Davoli, F., Bai, Y.: Construction of Simulation or Probabilistic Inference in uncertain and Dynamic Networks Based on Bayesian Networks. In: Proceedings of the International Conference on ITS Telecommunications, pp. 983–986 (2006)
30. Ding, J.: Probabilistic Fault Management in Distributed Systems. Ph. D. dissertation, FernUniversität in Hagen, Germany (2008)
31. Barco, R.: Bayesian modeling of fault diagnosis in mobile communication networks. Ph. D. dissertation, Universidad de Málaga, Spain (2007)
32. Cheng, L., Qiu, X., Meng, L., Qiao, Y., Li, Z.: Probabilistic Fault Diagnosis for IT Services in Noisy and Dynamic Environments. In: Proceedings IEEE INM 2009, pp. 149–156. Hofstra University, Long Island (2009)

33. Barco, R., Guerrero, R., Hylander, G., Nielsen, L., Partanen, M., Patel, S.: Automated troubleshooting of a mobile communication network using Bayesian networks. In: Proceedings of the IEEE International Workshop on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden, pp. 606–610 (2002)
34. Lee, G.J.: CAPRI: A Common Architecture for Distributed Probabilistic Internet Fault Diagnosis. Ph. D. dissertation, CSAIL-MIT, Cambridge, MA, USA (2007)
35. Wooldridge, M.: An Introduction to Multi Agent Systems, 2nd edn. John Wiley & Sons, Chichester (2009)
36. Xiang, Y.: Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach. Cambridge University Press, Cambridge (2002)
37. Pan, R., Peng, Y., Ding, Z.: Belief Update in Bayesian Networks Using Uncertain Evidence. In: 18th IEEE International Conference on Tools with Artificial Intelligence, pp. 441–444 (2006)
38. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Technical Report KSL-91-02, Knowledge Systems Laboratory. Medical Computer Science. Stanford University School of Medicine, Stanford, CA 94305-5479 (1993)
39. Friedman, N., Geiger, D., Godlzsmit, M.: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997)
40. Hastie, T., Tibshirani, R., Friedman, J.: The EM algorithm. In: The Elements of Statistical Learning, pp. 236–243. Springer, New York (2001)
41. TR-069 CPE WAN Management Protocol, `http://www.broadband-forum.org/technical/download/TR-069.pdf` (last visited July 2009)
42. JADE (Java Agent DEvelopment Framework), `http://jade.tilab.com` (last visited March 2010)
43. WADE (Workflows and Agents Development Environment), `http://jade.tilab.com/wade` (last visited March 2010)
44. The Foundation of Intelligent Agents, `http://www.fipa.org/` (last visited March 2010)
45. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. John Wiley & Sons, Chichester (2007)
46. Apache Felix, `http://felix.apache.org/site/index.html` (last visited March 2010)
47. OSGi, `http://www.osgi.org` (last visited March 2010)
48. Da Costa, P.: Bayesian semantics for the Semantic Web. George Mason University Fairfax, VA (2005)
49. JADE Ontology bean generator, `http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator` (last visited March 2010)
50. GeNIe, `http://genie.sis.pitt.edu/` (last visited March 2010)
51. Samiam project. Automated Reasoning Group at UCLA, `http://reasoning.cs.ucla.edu/samiam` (last visited March 2010)
52. SMILE website. Decision Systems Laboratory, Department of Information Science and Telecommunications and the Intelligent Systems Program, University of Pittsburgh, `http://genie.sis.pitt.edu`

A Multi-lane Double Auction for Economic-Based Service Management in the Cloud

Xavier Vilajosana, Daniel Lázaro, Angel. A. Juan, and Joan Manuel Marquès

Abstract. Economic models have shown their suitability to allocate resources efficiently, considering an unbalanced supply and demand. As the use of the Cloud is extending, a numerous set of distributed resource allocation frameworks have been developed to attain efficient resource management while keeping the scalability of the infrastructure. However, those frameworks make use of either simple double auction mechanisms or complex approximations to the NP-complete problem of the combinatorial auction. The problem of those mechanisms is that of its generality, that is, they have not been specially designed for the trading of time-leased computational resources. In this paper we present a novel variant of the double auction that has been specially adapted to trade time-differentiated items as Cloud services can be considered. The paper presents the data structures, algorithms and architecture of the economic mechanism as well as it presents the evaluation of the mechanism through simulation. Simulated results are compared with the main double auction implementations found in the literature. The paper constitutes an approach to improve efficiency of service management and allocation in the Cloud from the point of view of the economic model and not from architectural aspects addressed by most of the contributions found in the literature.

1 Introduction

Auction mechanisms have been used to allocate resources in computational environments [Lai et al(2005)Lai, Rasmusson, Adar, Zhang, and Huberman, Eymann et al(2003)Eymann, Reinicke, Ardaiz, Artigas, Freitag, and Navarro, Buyya and Venugopal(2004), Neumann et al(2007)Neumann, Stößer, Anandasivam, and Borissov, Consortium(2008), Haussheer and Stiller(2005)]. Most of the used

Xavier Vilajosana · Daniel Lázaro · Angel. A. Juan · Joan Manuel Marquès
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
e-mail: {xvilajosana,dlazaroi,ajuanp,jmarquesp}@uoc.edu

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 117–148.
springerlink.com

economic institutions are the Double Auction (DA) or the Combinatorial Auction (CA). The DA is a simple but powerful mechanism to allocate a single type of item amongst multiple buyers and multiple sellers. Combinatorial Auctions [Liu and He(2007), Radhanikanth and Narahari(2009), Schnizler and Neumann(2007)], in contrast, are NP-complete mechanisms that can deal with bundled items and substitute preferences. Both mechanism have been used with success in the allocation of resources in the Grid, however neither CAs nor DAs have been improved to fit to the nature of the services in the Cloud (heterogeneous and strictly time-leased). As already stated, DAs can only trade a single type of resource per instance, even multiple units, this constitutes a limitation for the allocation of services, mainly heterogeneous and differentiated by time. On the other hand, Combinatorial Auctions are not able to allocate high quantities of items due to their computational costs. Our work presents a novel approach of the Double Auction that improves the usability of Double Auctions as mechanism to allocate time-differentiated resources and services. Our approach, the Multi-Lane Double Auction (MLDA) provides the means by which multiple time differentiated items can be allocated by a single instance of the mechanisms. Besides it enables the allocation of substitute items approximating its results to the results obtained by CAs. The MLDA can be set in between both existing approaches constituting a useful alternative. It fills the gap between DAs and CAs and constitutes a good candidate when dealing with the allocation of heterogeneous computational resources/services. The paper presents the data structures, algorithms and architecture of the economic mechanism as well as it presents the evaluation of the mechanism through simulation.

2 The Double Auction

Double auctions determine the winners set amongst multiple buyers and sellers when offering to buy or sell single items. To clear the auction, the M th and (M + 1)st prices are computed, where M is the number of sell bids. It is assumed that a total order can be imposed on all the bids. This is commonly accomplished using price as the principal priority measure, and using bid quantity or bid placement time to break ties. Conceptually, finding the M th and (M + 1)st bids is simply a matter of sorting the bids in descending order, and identifying the Mth and (M + 1)st items in the list. The prices between the M th and (M + 1)st bids (inclusively) represent the range of prices for which supply balances demand. At prices in the range, the number of buyers willing to buy at that price equals the number of sellers willing to sell, with the caveat that when M th= (M + 1)st, one side or the other may have some participants who lose on tie-breaking criteria. It is important to note that this process of identifying the equilibrium price range works regardless of the relative position of the buyers and sellers in the list. The k-Double Auction computes a clearing price that is a ratio of the two boundary prices. Furthermore, the M th and (M + 1)st prices delineate the set of currently winning bids, referred to as the transaction set. Again, modulo ties at the boundaries, buyers at or above the M th-price would purchase an item if the auction cleared, and sellers at or below the (M + 1)st price would sell an item. It follows that the M th-price and (M + 1)st-price constitute exactly the information that is typically provided to participants in the form of price quotes. The M th-price is the ask quote and informs a potential buyer of the minimum that she would have to offer to be certain to enter the current transaction set. Symmetrically, the bid quote, equal to the (M + 1)st-price, informs a potential seller the maximum that he would be able to offer to become a current winner. Figure 1 exemplifies the above described procedure.

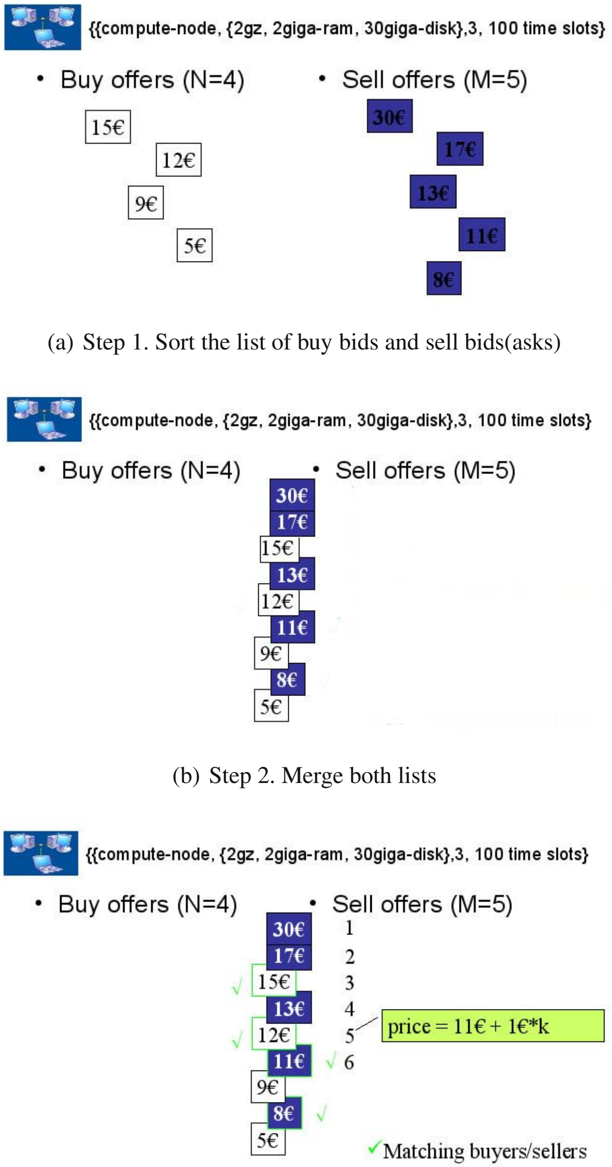
3 MLDA Functionalities

The Multi-Lane Double Auction (MLDA) has to support multiple time-differentiated items per instance while keeping the general functionalities offered by an auction. Since multiple time-differentited items must be supported, the structure of the MLDA has been designed as a set of lanes where each lane represents an item. The main operations offered by the MLDA are:

- **Insert/Remove Bid.** When a new bid is received and the auction system verifies that it satisfies whatever bidding rules exist. It must be inserted into the auctions data structures for its corresponding lane. Similarly, when a bid is withdrawn, it must be removed from the data structures. Inserted bids can be precise which means that they can only be inserted in a specific lane. Imprecise bids are those that can be placed in more than one lane.
- **Compute Quote.** The auction will generate price quote information for a lane .
- **Clear.** At designated times, the auction will compute exchanges between the buyers and sellers, notify the participants, and remove the winning bids from the data structures.

4 MLDA Operations

The MLDA algorithm has been developed following the same idea as the general double auction mechanism presented in the previous section. Bids and asks are organised in lanes, and for each lane the general double auction data structures are maintained. For each lane the MLDA keeps four structures to maintain bids sorted and another one global structure to buffer bids representing substitute preferences (henceforth *imprecise bids*). In the literature many different data structures can be found for such an aim: Heaps, Internal Path Trees, AVL or sorted lists. However, for the description of the MLDA algorithm it will be considered the use of sorted lists as they keep concepts simple and it does not affect the general functional behaviour of the algorithm. As indicated, each lane is represented by four sorted lists to store winning bids (Bin) , winning asks (Sin), losing bids (Bout) and losing asks (Sout). Bin is sorted in ascending order so that the lowest winning bid is the head of the list while Bout is sorted in descending order to keep the highest losing bid at the head of the list. Sin is sorted in descending order, keeping the highest winning ask at the head and Sout is sorted in an ascending order to keep the lowest losing bid at its



(a) Step 1. Sort the list of buy bids and sell bids(asks)

(b) Step 2. Merge both lists

(c) Step 3. Find the price. It is the Mth bid in descending order. Mth represents the number of sell bids received.

Fig. 1 Steps to compute winners in a double auction

head. By construction the MLDA keeps the Social Welfare maximum. This is the invariant of the insertion algorithm and is a key point for the economical efficiency of the MLDA. Social Welfare is a measure of wellness that permits to weight up how resources are distributed taking into account their need.

The algorithm does not differentiate between precise or imprecise bids, as a bid can be placed in one, two, three or more lanes, so the choice to where the bid will be placed will be guided by the possible lanes where the bid can be inserted. One important detail to consider and a fundamental issue to be solved by our algorithms, concerns to how imprecision is handled. Note that if only precise bids (without indicating substitute items) are inserted in the MLDA, it behaves as a set of independent auctions since no bids are suitable to win in more than one lane. So, the simplest case for a MLDA occurs when all bids are precise. In this case lanes can be cleared independently because no bid in a lane can displace a bid in another lane. Contrarily, when there are imprecise bids (substitute preferences) that can be inserted in more than one lane the efficiency obtained by the auction will be directly affected by the placement of the imprecise bids. Thus, when an imprecise bid B' is placed in a lane, the bid B' may prevent a precise bid B" from winning in that lane. In order to avoid any possible inefficiency, an imprecise bid may win only in the lane where the social welfare is maximised.

4.1 Bid Insertion

When a bid B is inserted in a MLDA, the list of possible lanes L where the bid can be placed is given. B can only affect the social welfare of any of the lanes in L. Many different situations can happen:

- B can displace a winning bid in any of the lanes in L
- B can make a current losing ask in any of the lanes in L be promoted.
- B cannot win in any of the lanes in L.

The condition to be maintained is that the Social Welfare is kept at a maximum level, so that the choice of any of those situations is given by the condition that maximises the current social welfare. So the question now is how to calculate the social welfare without having to compute it for every lane. In our algorithms we keep pointers to the current Lowest Losing Ask (LLA), the Lowest Winning Bid(LWB) and the Highest Losing Bid(HLB). Pointers are updated each time a bid or ask is inserted. Using these pointers, we can easily find the lane where the social welfare is maximised by inserting B. If a bid has to be displaced and substituted by B, the bid to be displaced will be the Lowest Winning Bid, because we want to maximise the welfare and the LWB is the worst bid that can be displaced. A displaced Bid, instead of being inserted in the losers list directly is kept in a buffer that we call the PendingLosingBids queue. Contrarily, when the situation that maximises social welfare is the one that corresponds to a promotion of a losing ask, the best choice will be the Lowest Losing Ask because there is not another ask, that if promoted, the social welfare can be improved. The pointer to the Highest Losing Bid is used to determine whether B can be discarded directly and inserted to the PendingLosingBids queue. As long as bids arrive, and there are no asks, they are directly inserted in the PendingLosingBids queue. This queue acts as a buffer and maintains bids sorted in a descending order. As we will see later the ask insertion will take advantage of this queue.

4.2 Ask Insertion

We considered that asks cannot be imprecise because it does not make sense for a seller to offer imprecise time-specified services. A seller will always indicate the specific service that it is selling including its specific time slot. The ask insertion algorithm also has to maintain the invariant, that is, keep the social welfare at a maximum level at each ask insertion. When an ask S is inserted in a lane several things need to be checked. If S is higher than the current Lowest Losing Bid, there is nothing to do and S has to be inserted in the Sout list in its lane. In any other situation S has a chance to be a winner. To make S win a currently losing bid B that can be placed in L has to be found. Once B is found ,it can be be promoted as winning bid in L and matched with S. Due to our invariant, the selected bid has to be such a bid that keeps the social welfare at a maximum level. This condition holds when the selected bid is the Highest Losing Bid that can be placed at L. The HLB that can be placed at L can be found either in the PendingLosingBids queue or in the Bout list in a lane. The algorithm selects the highest out of the possible bids. Even in that situation another condition has to be checked. It can happen that a current winning ask S' in a lane L' is higher than S and in L' there is a winning bid B' that can be moved to L. In this case, the social welfare would be improved by displacing B' to L and removing S' from the Sin in L'. To determine the best choice the following condition is checked:

whenever $B - S <= B' - S' + B - S$ it is better to promote the Highest Losing Bid that can be inserted at L. Otherwise, it is better to displace B' from L' to L and remove S' from the Sin in L'. Finally in the case that there are no suitable bids to be inserted/displaced at L, S is directly inserted in Sout in L.

4.3 PendingLosingBids Queue

The PendingLosingBids queue is a data structure that keeps bids organised in lanes. For each lane a decreasing sorted list is kept. Whenever a bid is inserted in the pending queue, a pointer to the bid is inserted in each lane where the bid can be placed. It offers functionalities to get the maximum bid out of a set of lanes. As introduced before, the PendingLosingBids queue is used to keep bids that at insertion time are not able to win or have been discarded. As asks arrive bids are removed from PendingLosingBids queue. Whenever no more asks arrive, bids are kept in the queue and considered to be losing bids.

```
Data: A Bid Bnew and $P_{lanes}$ the list of lanes where Bnew can be inserted
Result: A bid is inserted in its corresponding lane or in the pending queue
begin
    LWBordered $\leftarrow$ List of the LWB ordered increasing
    LLAordered $\leftarrow$ List of the LLA ordered increasing
    HLBordered $\leftarrow$ List of the HLB ordered decreasing
    initialize (LWBmin,LLAmin,HLBmax)
    if LWBordered is $\emptyset \wedge$ LLAordered is $\emptyset \wedge$ HLBordered is $\emptyset$ then
        insertPendingSortedLosingBids (bnew)
        return
    if LLAordered is $\emptyset \wedge$ Bnew $\geq$ HLBmax then
        if Bnew $>$ LWBmin then
            displaceLWB (Bnew, LWBmin)
        else
            insertPendingSortedLosingBids (Bnew)
    else if $\neg$ LLAordered is $\emptyset \wedge$ Bnew $\geq$ HLBmax then
        if LLAmin $\geq$ Bnew $\geq$ LWBmin then
            displaceLWB (Bnew, LWBmin)
        else if  LLAmin $\geq$ LWBmin $\geq$ Bnew then
            insertPendingSortedLosingBids (Bnew)
        else if LWBmin $\geq$ Bnew $\geq$ LLAmin then
            promoteLLA (Bnew, LLAmin)
        else if LWBmin $\geq$ LLAmin $\geq$ Bnew then
            insertPendingSortedLosingBids (Bnew)
        else if Bnew $\geq$ LLAmin $\geq$ LWBmin then
            displaceLWB (Bnew, LWBmin)
        else if Bnew $\geq$ LWBmin $\geq$ LLAmin then
            promoteLLA (Bnew, LLAmin)
        else
            return
    else if  $\neg$LLAordered is $\emptyset \wedge$ Bnew $\leq$ HLBmax then
        insertPendingSortedLosingBids (Bnew)
    else if LLAordered is $\emptyset \wedge$ Bnew $\leq$ HLBmax then
        insertPendingSortedLosingBids (Bnew)
    else
        return
end
```

Algorithm 1: Bid Insertion algorithm

```
Data: Snew the ask to be inserted in L the target lane.
Result: Inserts and ask.
begin
    if LLA (l) is $\emptyset \vee$ Snew $<$ LLA (l) then
        checkIfCanBePromoted (Snew, L)
    else if Snew $\geq$ LLA (l) then
        insertSout (Snew,L)
    else
        return
    end
end
```

Algorithm 2: Ask insertion algorithm.

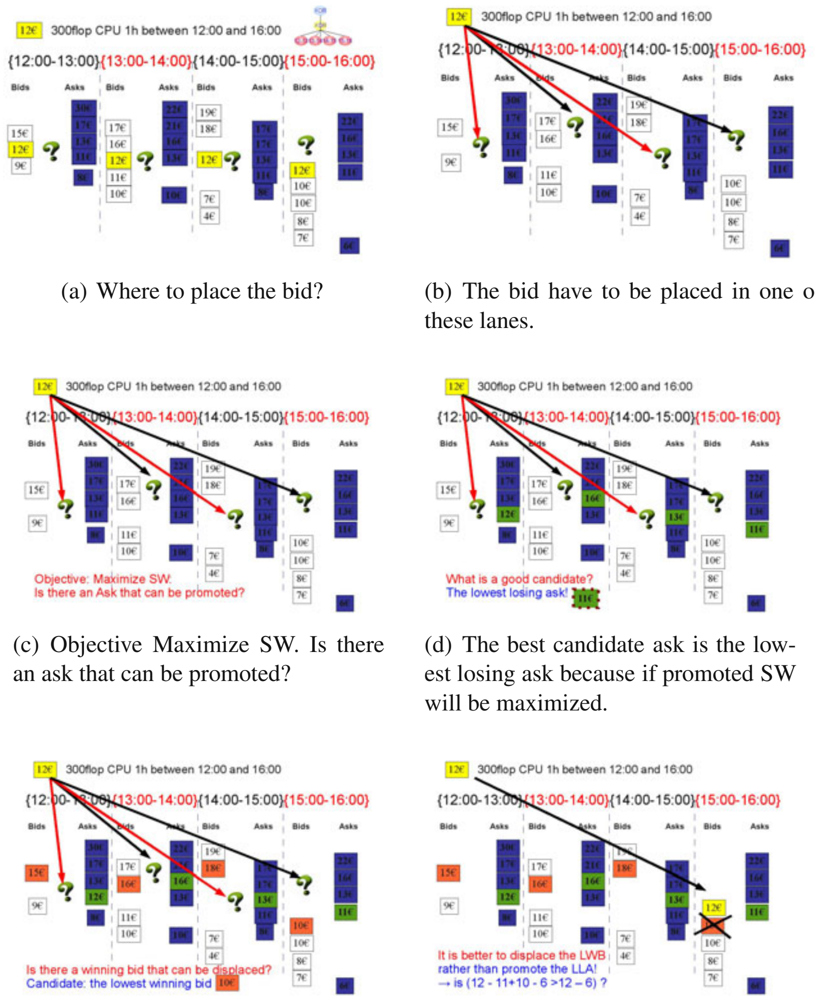
4.4 MLDA Clear and Quotes

The clearing operation is simple and straightforward. Clearing can be done, sequentially or in parallel because dependencies amongst lanes have been removed at insertion time. Thus, the clearing process matches highest bids with lowest asks by just traversing Sin and Bin. Price quotes are offered by every lane and are also easy to find, the lowest winning bid is the Bid quote and is given by the head of Bin that also corresponds to the Mth price as stated by the literature. (M+1)st price corresponds to the ask quote and is given by the head of Sin.

5 Implementation and Experiments

MLDA has been implemented and a set of experiments have been carried out. The aim of the simulation was twofold: first validate that MLDA provides optimal[1] efficiency and second compare its computational efficiency with multiple instances of single item double auctions. To carry out the experiments a set of data sets have been generated. Several distribution functions have been used to generate random data. The distribution functions used were derived from several experiments found in the literature [Phelps(2007), Phelps(2006), Mills and Dabrowski(2008)]. Uniform distribution of ask prices are motivated by the assumption that costs of services are also uniformly distributed. Bid prices have been generated using different distribution functions, Binomial distribution and Uniform distribution. Furthermore, bids and asks were also distributed across time slots following different distribution functions. Distributing bids and asks across different time slots using different distribution functions enabled us to experiment the effects of non-uniform supply/demand across time slots. Every experiment describes the reason for the distribution function used and the aim of the intended evaluation.

[1] There is no other allocation that improves the obtained social welfare.


12€
300flop CPU 1h between 12:00 and 16:00
{12:00-13:00}{13:00-14:00}{14:00-15:00}{15:00-16:00}
Bids
Asks
Objective: Maximize SW.
Is there an Ask that can be promoted?
What is a good candidate?
The lowest losing ask!
11€
Is there a winning bid that can be displaced?
Candidate: the lowest winning bid
10€
It is better to displace the LWB
rather than promote the LLA!
→ is (12 - 11+10 - 6 >12 – 6) ?

(a) Where to place the bid?

(b) The bid have to be placed in one of these lanes.

(c) Objective Maximize SW. Is there an ask that can be promoted?

(d) The best candidate ask is the lowest losing ask because if promoted SW will be maximized.

(e) But.. if there is a currently winning bid that if displaced SW can be improved...

(f) For this example is better to displace a winning bid.

Fig. 2 Process of inserting a bid into a MLDA instance. Selection of the best candidate lane to place the bid.

```
Data: Lanes a Map of lists representing the Multi-Lane catalog of bids
Result: Returns a Map of lists. Each lists contains the pairs of matched bids
        and asks.
begin
    Map $\kappa \leftarrow \emptyset$
    for $l \in Lanes$ do
        $\alpha \leftarrow emptyset$
        $\alpha \leftarrow$ getMatchings (l)
        put ($\kappa$,$\alpha$)
    end
    return $\kappa$
end
```

Algorithm 3: Clear algorithm.

5.1 Experiment A: Economical Efficiency

The experiment aimed to evaluate the economical efficiency obtained by the MLDA. Economical efficiency has been defined as the social welfare that the mechanism provides given a certain input. Social welfare has been computed as:

$$SW = \sum(Bids) - \sum(Asks)$$

5.1.1 Experiment A Setting

In order to evaluate the economical efficiency of the MLDA we aimed to make a comparison with another well-known double auction, the k-Double Auction (k-DA). The JASA k-DA framework [Phelps(2006),Phelps(2007)] implementation has been used to make a comparison with our implementation of the MLDA. The JASA k-DA was based on the 4Heap Algorithm implementation presented by Bao et. al [Bao and Wurman(2003)]. Experiments where conducted in a dual core T9500, 2.5GHz with 4Gb of Memory.

5.1.2 Experiment A Description

The following tables describe the experiments carried out.Table 1 summarizes our first set of experiments with MLDA. The experiment consisted of the generation of a set of 1000 bids and 600 asks[2]. Several auction instances were created, one MLDA auction instance for four lanes, and three sets of four 4HeapDoubleAuction

[2] The amount of bids and asks has been determined after several experimentation with different quantities of bids and asks, starting from 10 bids and 5 asks to 3000 bids and 1500 asks. 1000 bids and 600 asks have been considered a significative amount to evaluate MLDA. Of course, the choice have also been corroborated by other related work. Phelps thesis experiments with bid and asks sets of 30 to 1000 units. Mill and Dabrowski present different experiments using between 250 and 5500 processor requirements. In their homogeneous experiment, buyers required 500 processors and sellers offered 500 processors.

Table 1 Experiment A.1 setting

Experiment A.1				
Attribute	**MLDA**	**4HDARR**	**4HDAU**	**4HDAN**
Repetitions	100	100	100	100
Lanes	4	4 instances	4 instances	4 instances
Bids	1000 (all inserted at the MLDA instance)	1000 (distributed round robin at lanes)	1000 (Uniformly distributed at lanes)	1000 (Following a Binomial PDF with n=3 and p=0,4)
Asks	600 (Uniformly distributed amongst lanes)	600 (Uniformly distributed amongst lanes)	600 (Uniformly distributed amongst lanes)	600 (Uniformly distributed amongst lanes)

instances from JASA framework [Phelps(2006),Phelps(2007)]. Each set was referred as 4HDARR, 4HDAU and 4HDAN respectively.

The experiment marked all the bids for all the lanes (imprecise bids) of the MLDA, so as to indicate that the bids were for substitutable items. Afterwards all the asks were inserted and uniformly distributed amongst the lanes. The time of computation was measured. The computation time for all the experiments measured the time taken to initialise the instance of the auction, the time taken to insert all the bids and all the asks and then finally the time to clear the auction. Furthermore, the social welfare and the number of matches where computed at the finalisation of the experiment.

For the cases of 4HDARR, 4HDAU and 4HDAN, the same experiment was carried out. For 4HDARR, bids where inserted in a round robin fashion in each lane instead of being described as substitutable for all lanes as in the case of MLDA. This means that Bid 1 was placed in 4Heap Auction instance representing lane one, Bid 2 was placed in the 4Heap Auction instance representing lane 2, Bid 3 in the auction representing the 3rd and so on... The asks were inserted following a uniform distribution amongst the lanes. The experiment measured the time taken to compute the initialisation of the four instances of the 4HeapAuction, as well as the time taken to insert the bids and asks and clear the auction. 4HDAU and 4HDAN behaved accordingly but with the difference that bids where inserted following a uniform distribution and a binomial distribution respectively.

For the four experiments the same set of bids and asks where used in order to avoid divergences due to randomisation. The experiments were repeated 100 times, re-generating the bids and asks in each experiment.

Table 2 presents the setting for the second experiment. Experiment A.2 set the distribution of asks following a Binomial PDF with n=3 (the number of lanes (0 to 3)) and p=0.24 to centre it near to lane 1. Binomial distribution was used since we wanted to evaluate a distribution of asks where not all lanes had the same probability but some of them were preferred amongst others.

Table 2 Experiment A.2 setting

Experiment A.2				
Attribute	**MLDA**	**4HDARR**	**4HDAU**	**4HDAN**
Repetitions	100	100	100	100
Lanes	4	4 instances	4 instances	4 instances
Bids	1000 (all inserted at the MLDA instance)	1000 (distributed round robin at lanes)	1000 (Uniformly distributed at lanes)	1000 (Following a Binomial PDF with n=3 and p=0,4)
Asks	600 (Following a Binomial PDF with n=3 and p=0,24)	600 (Following a Binomial PDF with n=3 and p=0,24)	600 (Following a Binomial PDF with n=3 and p=0,24)	600 (Following a Binomial PDF with n=3 and p=0,24)

5.2 Results Analysis

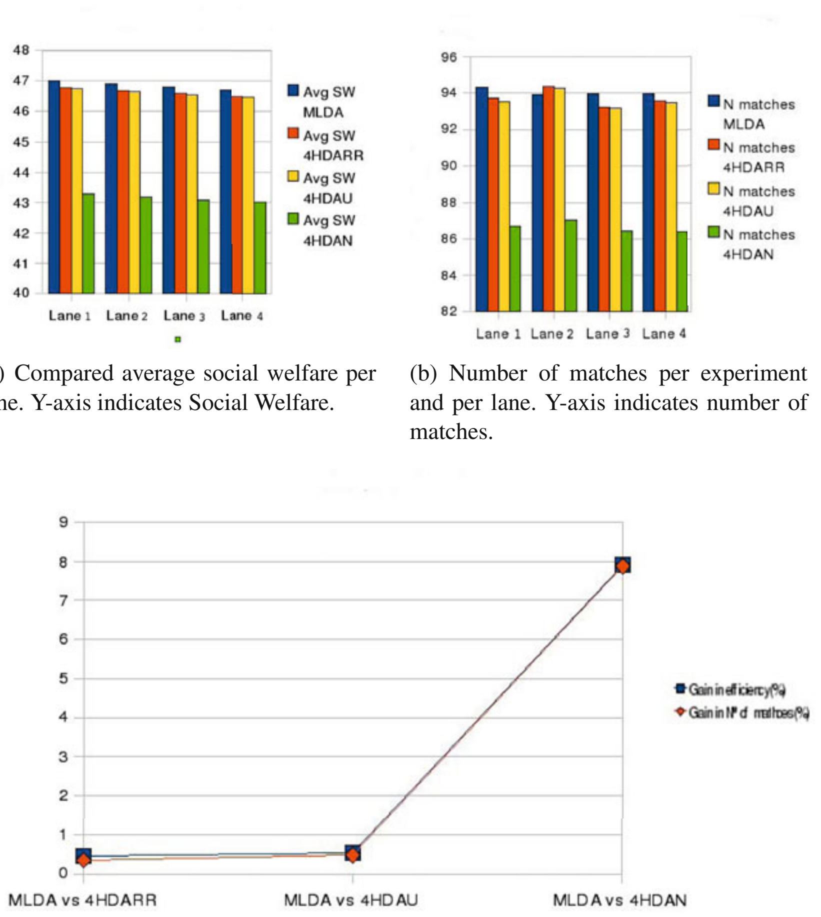
5.2.1 Experiment A.1 Results

Experiment A.1 aimed to evaluate the Social Welfare obtained by MLDA in the setting described above. By construction MLDA keeps social welfare optimal[3] so the expected results were that it achieves the best social welfare amongst other auctions. As can be seen in Figure 3.a the average social welfare amongst the 500 experiments in each lane is the highest for MLDA. 4HDARR and 4HDAU achieve close to MLDA social welfare, since the bids and asks are distributed proportionally in each lane. However, the benefit of MLDA is that the bids are always placed in the best option when 4HDARR and 4HDAU are restricted to placing them in one specific lane. 4HDAN achieves a worse social welfare due to an imbalance in the distribution between bids and asks.

The same expected results were achieved when considering the number of matches. As can be seen in Figure 3.b MLDA achieved the highest number of matches, either per lane or in total. For the 4HDAN, we can see that the lane with highest probability achieves the highest number of matches, but it still obtains the worst results.

Figure 3.c compares MLDA with 4HDA. The Figure presents the gain in % of MLDA when compared to 4HDARR,4HDAU and 4HDAN. The compared metrics have been economical efficiency, i.e. improvement in the obtained social welfare and number of matches provided. MLDA is slightly better, 0.5%, than 4HDARR and 4HDAU when asks are distributed following a uniform distribution amongst the lanes. In these cases, the improvement obtained by MLDA is not very significant due to the distribution of bids in 4HDARR and 4HDAU, which places bids almost uniformly across the lanes. As the asks and bids were placed following the same type of distribution, the number of matches per lane in 4HDARR and 4HDAU are

[3] See Section 4 for the description of the algorithm.



(a) Compared average social welfare per lane. Y-axis indicates Social Welfare.

(b) Number of matches per experiment and per lane. Y-axis indicates number of matches.

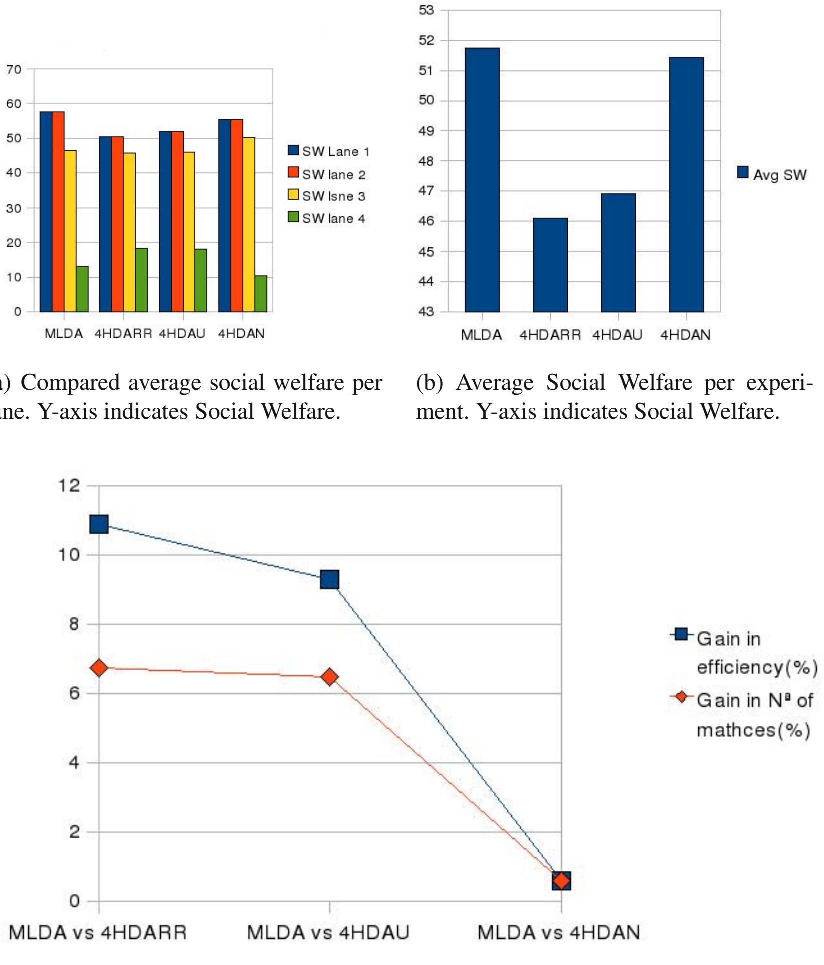
(c) MLDA compared to 4HDA. Y-axis indicates improvement in %.

Fig. 3 Results of the experiment A.1

close to the allocation obtained by MLDA. However, 4HDAN behaves poorly due to the distribution of the bids being MLDA at least 8% better than 4HDAN. This of course can be attributed to the imbalance between the distribution of the asks and bids that leaves lanes 1 and 4 with a lower number of bids, which reduces the overall welfare and number of allocations. In this Figure a direct relationship can also be seen between economic efficiency and the number of allocations. It can be deduced that the gain in efficiency is directly proportional to the gain in the number of allocations.

5.3 Experiment A.2 Results

Like experiment A.1, experiment A.2 aimed to confirm that the MLDA achieves the best Social Welfare when compared to multiple auction instances. This second experiment distributed the asks following a binomial distribution as previously described. The reason for such distribution is to see the effects of heterogeneous supply and demand distribution amongst the lanes.



(a) Compared average social welfare per lane. Y-axis indicates Social Welfare.

(b) Average Social Welfare per experiment. Y-axis indicates Social Welfare.

(c) MLDA compared to 4HDA. Y-axis indicates improvement in %.

Fig. 4 Results of the experiment A.2

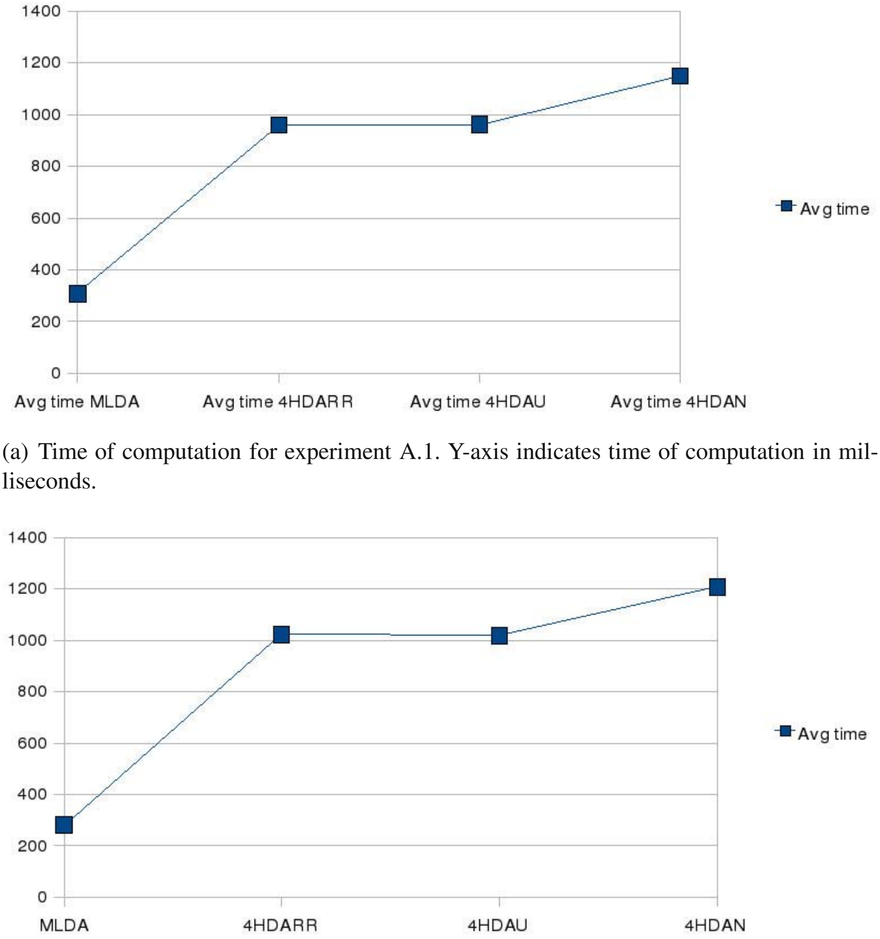
Figure 4.a shows that lanes 1 and 2 achieved higher social welfare compared to 3 and 4 and this, of course, is due to the distribution of the asks. Compared results show that MLDA gets the highest social welfare for lanes 1 and 2 while 4HDAN achieves the best social welfare in lane 3. Regarding average social welfare amongst lanes (see Figure 4.b), MLDA achieves the best social welfare followed by 4HDAN. 4HDAU and 4HDARR achieve the worst social welfare due to the binomial distribution of asks and their almost equiproportional distribution of bids.

Finally, Figure 4.c shows the gain in efficiency (in terms of social welfare) in % when comparing MLDA with other auctions. MLDA compared with 4HDARR shows that MLDA is 11% more efficient than 4HDARR, 8% more efficient than 4HDAU and even 1% more efficient than 4HDAN, when playing in its optimal situation.

5.4 Experiment B: Computational Efficiency

Experiment B aimed to evaluate the computational efficiency of MLDA. Experiments A.1 and A.2 have been used to obtain experimental data concerning the computational efficiency of the different auctions. Time of computation in milliseconds has been used to determine the computation efficiency of the auction. For each experiment, the same operations were measured. MLDA was measured once the data had been generated. Measurements started at MLDA instance creation and subsequent bid insertion. After the insertion of the asks and the clearing of the measurements ended. 4HDA (in any of their variants) were also measured after data generation. Namely, instance creation, bid and ask insertion and subsequent clearing were measured. Results show that 4HDA settings take almost four times longer to finish the computation. We attribute the extra time to initialise different instances, as well as its management. Moreover, we observe that when bids in MLDA are for a restricted set of lanes the time of computation is reduced due to diminutions of the search space of the MLDA algorithm (i.e. as more bids are restricted to one lane, the lower the number of bid searches in other lanes, besides, as the number of bids for all lanes reduces, there is less search time in the overall search space). In addition, we want to point out the effects of the PendingLosingBids queue that keeps losing bids in it instead of inserting them into the Bout structures at each lane. This also shortens the time of computation for MLDA. Figure 5.a shows the time taken to carry out experiment A.1. It shows that MLDA is almost 4 times better than other 4HDA. The 4HDAN is the worst case, produced apparently by a higher number of bids in one instance in respect to others that produce a higher number of bid displacements.

Figure 5.b shows similar results when asks are inserted following a binomial distribution.



(a) Time of computation for experiment A.1. Y-axis indicates time of computation in milliseconds.

(b) Time of computation for experiment A.2. Y-axis indicates time of computation in milliseconds.

Fig. 5 Results of the experiment B

5.5 Experiment C: Scale Sensibility

The test aimed to analyse the scalability in number of lanes of the MLDA. It is supposed to obtain a linear increment of computation time as the number of lanes increases linearly. Moreover, we aimed to prove that the relationship between the performance of MLDA and multiple instances of 4HDA are maintained.

The experiment has been defined as follow:

The results obtained can be seen in Figure 6. It can be seen that our expectations were met. MLDA keeps being 2 to 3 times faster as multiple auction instances of

Table 3 Experiment C.1 setting

Experiment C.1				
Attribute	**MLDA**	**4HDARR**	**4HDAU**	**4HDAN**
Repetitions	100	100	100	100
Lanes	1, 5, 10, 15, 20, 24 lanes	1, 5, 10, 15, 20, 24 instances	1, 5, 10, 15, 20, 24 instances	1, 5, 10, 15, 20, 24 instances
Bids	1000	1000 (distributed round robin at lanes)	1000 (Uniformly distributed at lanes)	10000 (Following a Binomial PDF with n=3 and p=0,24)
Asks	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)

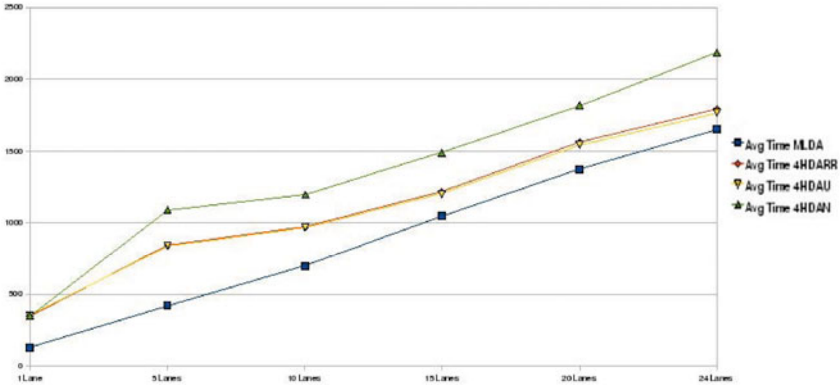
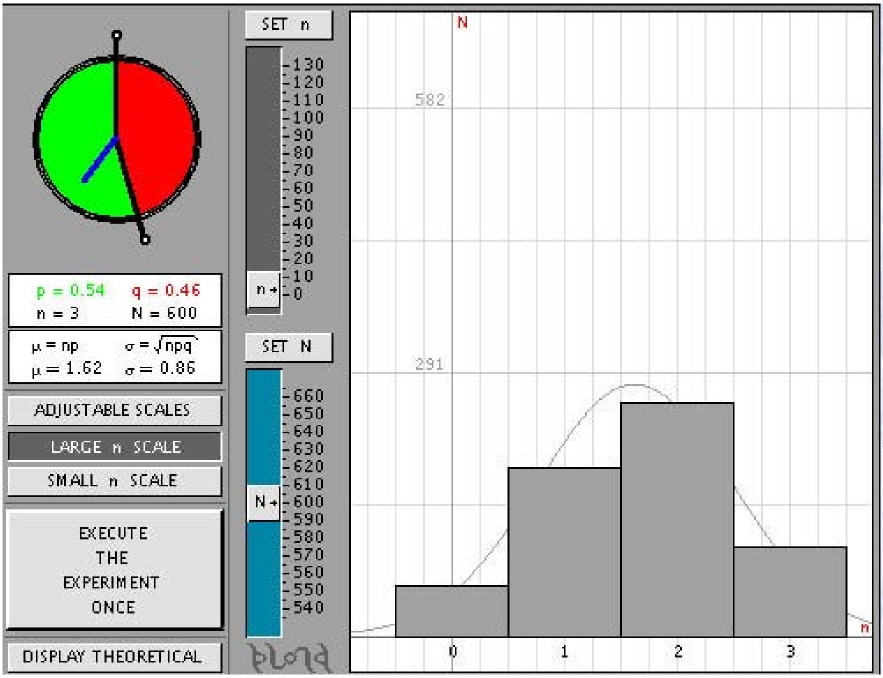


Fig. 6 Compared average execution time for different number of time slots (lanes). Y-axis indicates time of execution in milliseconds.

4HDA. We measured the scalability of MLDA in comparison to the others. One of the measures taken is the time increment between the different numbers of lanes for each experiment. On average, MLDA increases the time of computation to around 250 ms, being almost the same as in other 4HDA experiments. We conclude that there is a linear increment of time as the number of lanes increases.

5.6 Experiment D: Price per Time Slot

Up until now, the experiments did not consider the effects of bid prices in the final result of the auction. In this experiment we want to measure some of the effects of price distribution in the allocation provided by the auction, as well as the price



$p = 0.54$ $q = 0.46$
$n = 3$ $N = 600$

$\mu = np$ $\sigma = \sqrt{npq}$
$\mu = 1.62$ $\sigma = 0.86$

Fig. 7 Binomial Distribution for asks in experiment D.2

per time slot. Previous experiments distributed bids following different distribution functions amongst the lanes, however, prices were generated following a uniform distribution for both sellers and buyers. It is not clear right now, what is a correct distribution for bid prices in a market. It is not reasonable to assume that prices for bidders are distributed uniformly, but more probably bid prices can follow a Normal distribution or even a Pareto distribution for a specific lane. Contrarily for asks, it seems reasonable to assume that their prices follow a uniform distribution, due to the fact that costs can be assumed to be homogeneously distributed.

Experiment D.1 aims to evaluate the final price per time slot obtained by MLDA and 4HDA and verify that the results obtained by MLDA remain optimal. Furthermore, it aims to analyse the consequences of optimality[4] to the price per time slot. It is not clear whether the mechanisms that achieve best social welfare will achieve higher prices per time slot.

Moreover, prices per lane cannot be assumed to be uniformly distributed, but they should follow a long tailed distribution, such as a left shifted Normal distribution or a Pareto distribution. The reason for that is that almost all bidders want their services to be allocated as soon as possible, so as nearer the time, the higher their willingness to pay for the service. For that reason the experiment will place bids with higher prices close to the first offered time slot. Thus, the willingness of buyers

[4] In social welfare terms.

Table 4 Experiment D.1 setting

Experiment D.1				
Attribute	**MLDA**	**4HDARR**	**4HDAU**	**4HDAN**
Repetitions	100	100	100	100
Lanes	4 lanes	4 instances	4 instances	4 instances
Bids	1000 for all lanes with prices distributed in a N(0.5,0.2)	1000 distributed round robin at lanes and prices distributed in a N(0.5,0.2)	1000 Uniformly distributed at lanes and prices distributed in a N(0.5,0.2)	10000 Following a Binomial PDF with n=3 and p=0,24 and prices distributed in a N(0.5,0.2)
Asks	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)	600 (Following a Uniform distribution)

to pay more to obtain earlier services will be simulated. In this experiment prices will be determined following the k-pricing rule that computes a non-discriminatory price for all matches in the lane. The k value has been set to 0.5 to distribute welfare in an equitable way amongst buyers and sellers. Prices have been computed as:

$$p = k \times (pM + 1) + (1k) \times pM$$
$$s.t. 0 \leq k \leq 1$$

where the pMth price and pM+1st price are the price quotes for the lane. Experiment D.2 will discuss some of the results obtained by the experiment described in the following table:

Table 5 Experiment D.2 setting

Experiment D.2	
Attribute	**MLDA**
Repetitions	100
Lanes	4 lanes
Bids	500 for lanes 1 and 2 with prices distributed in a N(0.65,0.2) and 500 for lanes 3 and 4 with prices distributed in a N(0.5,0.2)
Asks	600 (Following a Binomial PDF with n=3 and p=0.54 distribution amongst lanes)

5.7 Experiment D.1 Results

As described in Table 4, the experiment set a Normal distribution of ask prices centred at 0.5 and with a standard deviation of 0.2, this makes prices appear normally distributed between 0 and 1. It seems more reasonable to assume that prices in a market are distributed following a non-homogeneous distribution where a high percentage of bidders express similar valuations for the time slot rather than a uniform distribution of prices between 0 and 1 as in previous experiments. Figure 8 shows the distribution of the average prices per time slot (per lane) achieved by MLDA and other experiments with 4HDA. In that configuration, asks have been distributed uniformly across lanes and this is important to understand the results obtained. MLDA, achieves almost a constant price per time slot at around 0.57 due to the capacity of MLDA to place bids in the lane where the social welfare is maximised and because of the uniform distribution of asks that makes that the best configuration consists of distributing bids homogeneously across lanes. 4HDARR and 4HDAU achieve similar results as MLDA with slight more variations due to their incapacity to adapt the demand to the offer in an optimal manner. Their prices per time slot are around 0.57 and 0.55. Worse results are obtained by 4HDAN, due to the way bids are distributed. Lane 1 achieved a very low price per time slot, due to the low demand received in that time slot. Contrarily, Lane 3 obtained a high price per time slot (0.7) that can be attributed to the higher demand for that time slot. We conclude that there is a direct relationship between the demand for a time slot and the final price achieved in that Lane when the offer is fixed.

On the right-hand of Figure 8 the average prices per time-slot paid by every transaction can be seen. 4HDAN achieves higher prices due to a lower number of matches. However, 4HDAN achieves the worst Social Welfare which confirms that higher prices per time slot does not indicate better mechanism efficiency, in fact higher prices are a result of supply and demand balance for each lane. Figure 9 shows the number of matches per lane obtained by the fourth experiments. As already stated, MLDA achieves the highest number of matches while 4HDAN the worst due to the distribution of bids and asks.

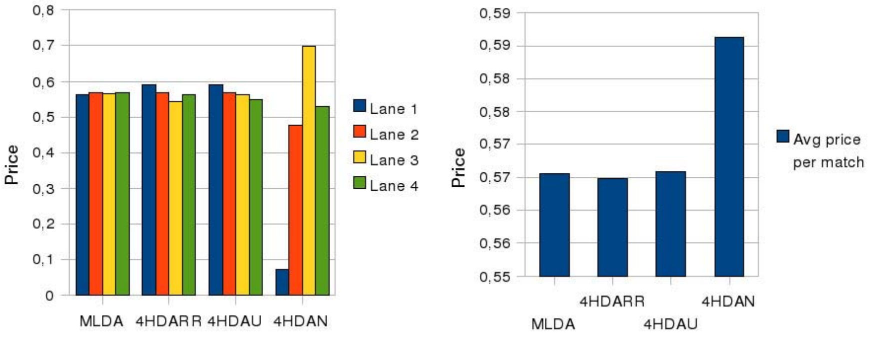


Fig. 8 Distribution of prices per lane and prices per time slot. Y-axis indicates price.

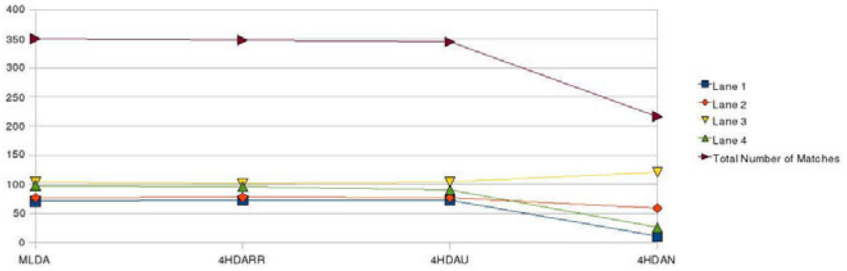


Fig. 9 Matches per lane and total number of matches. Y-axis indicates number of matches.

5.8 *Experiment D.2 Results*

As described in Table 5 the experiment aimed to price the first two time slots (lanes) higher to simulate the willingness of buyers to allocate services as soon as possible. Prices for lanes 1 and 2 have been generated following a Normal distribution with a mean of 0.65 and a standard deviation of 0.2. Prices for lanes 3 and 4 have been calculated using a Normal distribution with mean of 0.5 and standard deviation of 0.2. Asks where distributed non-uniformly across lanes (following a binomial distribution with n=3 and p=0.54), establishing a major number of asks for lanes 2 and 3.

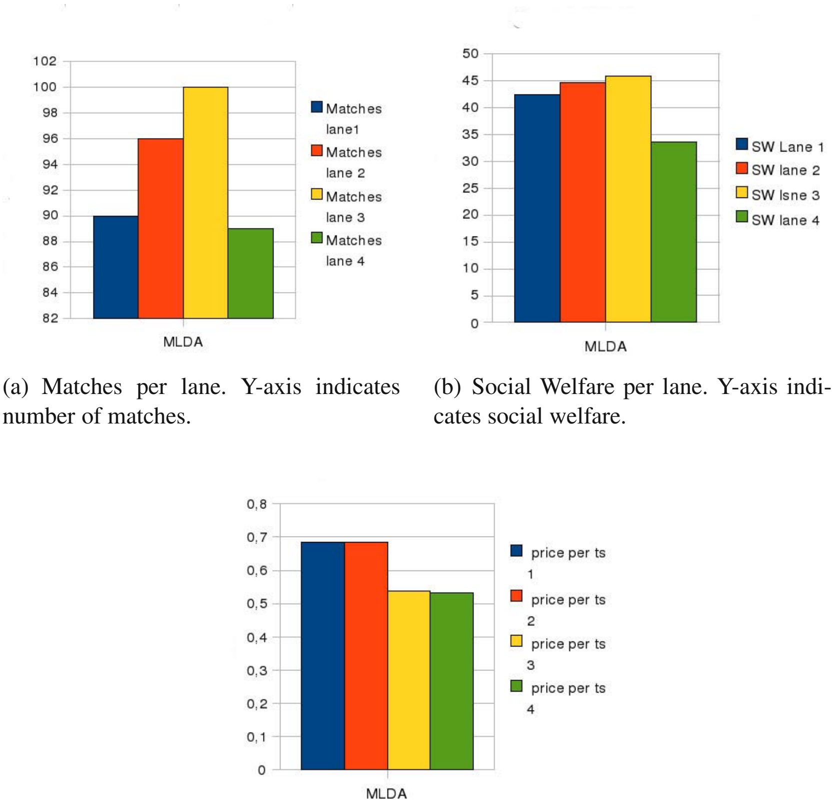
Figure 10.a shows the number of matches per lane obtained by the MLDA. Lane 3 is the lane that obtains more matches due to the distribution of the asks. Lane two obtains a lower amount of matches, even though prices are higher, due to a lower quantity of asks in that lane.

Figure 10.b shows the distribution of the social welfare generated by MLDA amongst lanes. The asks distribution guides the number of matches per lane and is the most significant factor for the final allocation provided by MLDA. Finally, the effects of higher prices in lanes 1 and 2 can be seen in Figure 10.c. Even though lane 2 has a higher number of matches, which should mean a lower price per time slot, it achieves a similar price per time slot to lane one that achieves the maximum due to a short offer and higher bid prices.

It can be concluded that asks distribution constrains the type of allocation provided by MLDA since asks can only win in the lane for where they have been submitted. It can also be pointed out that the higher the bid price, the higher the number of matches when asks have uniform prices.

5.9 *Experiment E: Memory Usage*

Experiment E aimed to evaluate the memory consumption of MLDA when compared to any of the 4HDA implementations used so far. The experiment aims to analyse the overall amount of memory used during the process of bids and asks insertions. It is clear that the amount of memory used will depend on the size of



(a) Matches per lane. Y-axis indicates number of matches.

(b) Social Welfare per lane. Y-axis indicates social welfare.

(c) Price per time slot. Y-axis indicates price.

Fig. 10 Results of the experiment D.2

the data structures used in the implementation. For this experiment is not important the total amount of memory used by both instances, but the relation between the amount of memory used. What is important to know is whether MLDA uses less, more or the same amount of memory than 4HDA, as well as the ratio of the difference. Table 6 summarises the experiment setting that consisted of 500 iterations of an experiment that inserted 1000 bids and 600 asks to an instance of every one of the evaluated auctions.

Memory usage has been measured during the insertion of the bids and asks. Measures were taken just before instantiating the auction and just after finishing the insertion of the last bid and ask. The system's garbage collector has been called before the first measurement and just after the last measurement. The amount of memory used has been calculated as a difference between the initial amount of memory and the final amount of memory.

Table 6 Experiment E.1 setting

Experiment E.1		
Attribute	**MLDA**	**4HDA**
Repetitions	500	500
Lanes	4 lanes	4 instances
Bids	1000 for all lanes	1000 round robin amongst lanes
Asks	600 (Following a Uniform distribution amongst lanes)	600 (Following a Uniform distribution amongst lanes)

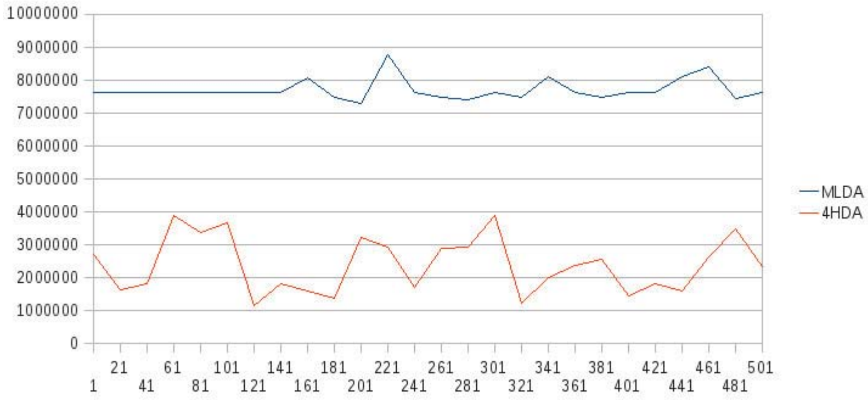
5.10 Experiment E Results

The results of the 500 experiments can be seen in Figure 11.a. MLDA spends nearly 3 times more memory than 4HDA in almost all experiments. Slight variations of memory usage at each experiment can be attributed to the runtime. These variations are more significant at 4HDA since four instances of auction objects are maintained. However, the general line is well defined and the relation is almost constant. Figure 11.b shows the ratio between MLDA and 4HDA. MLDA uses in average 3.23 times more memory than 4HDA, which is attributed to the PendingLosingBids queue that maintains a pointer for each bid to the lanes where a bid can be placed, and to the sorted lists used to maintain the different quotes across lanes.

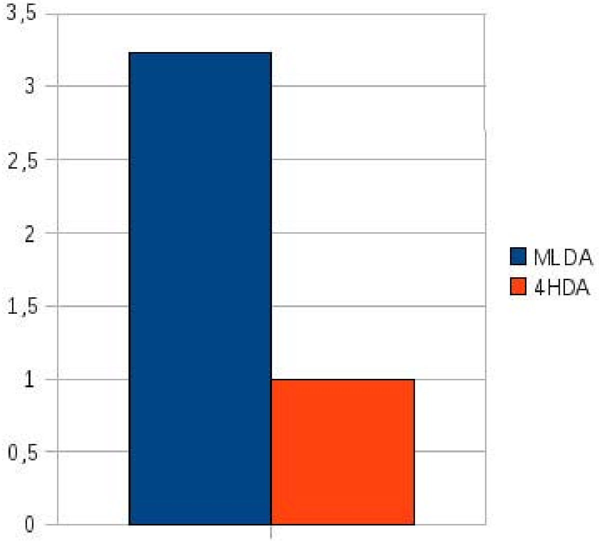
MLDA achieves better computational performance, as demonstrated in previous experiments, at the expense of using more memory. MLDA improves in computational efficiency two to three times to any 4HDA, while it used 3 more times of memory. However, MLDA provides the benefit of dealing with substitute preferences that any other 4HDA can deal with.

6 Related Work on Resource Allocation Frameworks

The term market mechanism is encountered in connection with problems of distributed resource allocation. In the context of markets it refers to a structure of economic organisation that helps to shape outcomes. Intuitively [Nisan and Ronen(2001)] a mechanism solves a problem by assuring that the required allocation occurs when agents choose their strategies to maximise their own utility. A mechanism also needs to ensure that the agent reported utilities are compatible with the algorithm implementing the mechanism. Economic mechanisms propose a procedure by which a set of resources may be distributed amongst the different participants and a scheme for pricing of the traded resources. The allocation is constrained by the preferences of the participants expressed in monetary terms. This section gathers the most significant auction based resource allocation frameworks found in the literature.



(a) Memory usage during 500 experiments. Y-axis indicates the amount of memory used in bytes.



(b) Compared memory usage between MLDA and 4HDA. Y-axis indicates the relation of the amount of memory required.

Fig. 11 Results of the experiment E

6.1 Auction Markets for Single Type of Resource

SPAWN [Waldspurger et al(1992)Waldspurger, Hogg, Huberman, Kephart, and Stornetta] was designed to tap into unused and wasted cycles in networked servers. Each participating server runs an auction process to trade the CPU time in fixed time-slices. Spawn uses a sealed bid second-price auction, known as the Vickrey auction. Vickrey auction is incentive compatibility, i.e. the best strategy that the bidders may practise is to reveal their true valuations. This system is not generalised to multiple resources and multiple resource units - considering time-slices as a resource unit; this implies an auction for each time-slice.

Placek et. al. [Placek and Buyya(2006)] present a trading platform for storage services. The platform implements a centralised storage exchange that implements a double-auction; the exchange accepts sealed offers from providers and consumers and periodically allocates trades by employing an algorithm that maximises surplus, that is, the difference between the consumer's price and the seller's cost. Double auctions are adapted to the trading of a single type of homogeneous resource. These have the benefit of reducing communication costs (single bids) and with suitable pricing policies are also incentive compatible.

6.2 Auction Markets for Multiple Types of Resources

The combinatorial auction model has received a lot of attention in recent years; to address trading multiple resource types in bundles; this has two implications: (i) prices are expressed for bundles and (ii) a bundle, if allocated, should be completely satisfied.

Chun et. al. [Chun et al(2006)Chun, Ng, Albrecht, Parkes, and Vahdat] present a resource discovery and allocation system where users may express preferences using a bidding language supporting XOR bids, and at most one of the preferences is allocated. Multiple resources may be requested to a central auction server that clears periodically. Resource requests are for fixed durations of time and users may specify the time ranges. A greedy algorithm clears the combinatorial auction. This algorithm privileges execution time over efficiency of allocation - bids are ordered by decreasing values where the value is obtained by dividing the bid price by the product of total number of resources and the duration of request.

Schwind et. al. [Schwind et al(2006)Schwind, Gujo, and Stockheim] present an iterative combinatorial auction that maximises seller revenues. Bids are presented as a two-dimensional matrix; one dimension represents the time in fixed time slots and the other dimension the resources (CPU, Disk, and network). The auction server executes periodically and invites bids from the participants. Shadow prices are calculated for individual items (resource) and the buyers are requested to iterate on their bids based on the current estimation of prices. The clearing algorithm is implemented as a linear program optimising the revenue. Prices are calculated using the approach presented by [Kwasnica et al(2005)Kwasnica, Ledyard, Porter, and DeMartini] âĂŞ the prices are the dual solution to the primal Linear Programming Problem (LP).

Schnizler and Neumann [Schnizler(2007)] present a multi-attribute combinatorial auction that maximises the surplus - the difference between the buyer's price and seller's cost. Resources are traded in fixed time-slots and buyers send XOR bids specifying the quality and the number of time-slots within a specified time range.

The Vickrey pricing policy is applied to provide incentive compatibility. Simulation results show that the allocation problem is computationally demanding but feasible in the case where the number of participants and bids are reasonably small.

Bellagio [AuYoung et al(2004)AuYoung, Chun, Snoeren, and Vahdat] is a market-based resource allocation system for federated distributed computing infrastructures. Users specify interest on resources in the form of combinatorial auction bids. Thereafter, a centralised auctioneer allocates resources and decides payments for users. The Bellagio architecture consists of resource discovery and resource market. For resource discovery of heterogeneous resources, Bellagio uses SWORD [Albrecht et al(2004)Albrecht, Patterson, and Vahdat]. For resource market, Bellagio uses a centralised auction system, in which users express resource preferences using a bidding language, and a periodic auction allocates resources to users. A bid for resource includes the sets of resources desired, processing duration and the amount of virtual currency which a user is willing to spend. The centralised auctioneer clears the bid every hour. The resource exchange in the current system is done through a virtual currency. The virtual currency is the amount of credit a site has, which is directly determined by the site's overall resource contribution to the federated system. Bellagio employs Share [Chun et al(2004)Chun, Ng, Albrecht, Parkes, and Vahdat] for resource allocation in order to support a combinatorial auction for heterogeneous resources. Share uses the threshold rule to determine payments. Once the payment amount of each winning bid has been determined by the threshold rule, the winning bidders receive resource capabilities after being charged the appropriate amount.

Even though the above approaches indicate the computational complexity of combinatorial auctions, they nevertheless are important demonstrators. Combinatorial auctions are important mechanisms for Grid resource markets since typically Grid applications need to allocate resources in bundles.

6.3 Proportional Sharing Markets for Divisible Resources

Market based proportional sharing models are one of the most popular approaches in problem-solving environments. Basically this approach consists of allocating users a percentage of the resource that is proportional to the amount of the bid submitted by the user. This may be considered as a fair model of allocation and is typically employed in cooperative environments employed in systems where resources are considered as divisible.

Tycoon [Kevin Lai and Fine(2004)] is a system designed for time-sharing networked nodes, such as in PlanetLab; an environment where users of resources are also providers of resources. Resource nodes execute an auction process to which users may send their bids. Tycoon implements a proportional sharing [Kelly(1997)] auction where resources are shared and each user is attributed a capacity proportional to its bid. Users are price-anticipating in that the ratio that they receive is a proportion of their bid over the sum of all bids for a given resource and users may anticipate the effect of their bids on the clearing price. Each user has a utility function; a weighted sum of the resource fraction that it receives from each node. Users bid to those nodes that maximise their utility. This system is intrinsically decentralised as the maximisation of the utility is done locally by each user. This system is appropriate for divisible usage of a CPU, an assumption that may not be acceptable to a wide range of applications.

6.4 Decentralised Markets for Single Type of Resources

Peer-to-peer auctioning has emerged as a new computing paradigm to decentralise the auction processes. Many systems address the decentralised auctioning issue for different reasons like those of scalability, fault-tolerance, redundancy, load distribution and autonomy, amongst others. With respect to scalability issues, most of the existing systems make use of a DHT structured overlay [Ratnasamy et al(2001)Ratnasamy, Francis, Handley, Karp, and Schenker, Stoica et al(2001)Stoica, Morris, Karger, Kaashoek, and Balakrishnan, Castro et al(2002)Castro, Druschel, Kermarrec, and Rowstron, Ghodsi(2006)].

PeerMart [Hausheer and Stiller(2005)] distributes brokering in an auction based allocation mechanism. Auctioneers, rather than being a single broker, are formed by a set of peers which synchronise to clear a double auction market. Consumers and providers are distributed in the Pastry overlay network [Castro et al(2002)Castro, Druschel, Kermarrec, and Rowstron]. Broker sets are formed by some nodes in the overlay and the double auction they implement clears continuously. For each allocation, brokers synchronise their information in order to determine who the winners are and to avoid malicious peers. Synchronisation is carried out by broadcasting the lowest selling requests, the highest buying bid and any matching bid to the rest of the brokers in the broker set. Decisions are taken by a simple majority. Broker set peers maintain a distributed shared state through broadcasting information and decisions are taken when all peers in the broker set have all the information.

Tamai et.al. [Tamai et al(2005)Tamai, Shibata, Yasumoto, and Ito] use CAN [Ratnasamy et al(2001)Ratnasamy, Francis, Handley, Karp, and Schenker] to build a market architecture. They propose distributing peers into different sub-regions and assigning a responsible broker for each of them. Sell requests and buying bids are sent to any known broker and they are forwarded until they reach the broker responsible for the offer. If the broker has a buying bid (sell requests) that matches the received sell request (buying bid) the allocation is made. However, if a buying bid and a sell request are received by different brokers respectively, they cannot be matched. To solve that problem, Tamai et.al. propose replicating buying bids and sell requests in multiple brokers. The replication introduces more communication amongst peers. When a replica matches a sell request with a buying bid it has to verify that the original bid (request) has not been matched by sending a message to the original replica. Once a buying bid and a sell request are matched, a replica deletion message has to be sent to all replicas.

As proof of this concept, in the paper by Despotovic et.al. [Despotovic et al(2004)Despotovic, Usunier, and Aberer], a simple approach is presented. The paper presents a mechanism for pricing and clearing continuous double auctions in a peer-to-peer system. The main feature is that consumers and providers broadcast bids for resources. Every buyer has the incentive to trade with the announcer of the lowest sell request that the buyer observed. Similarly, any seller would want to trade with the announcer of the highest observed bid. Prices in each trading operation are set to the average price between the bid and the sell request. Since peers do not have a global view of all the trading operations that occur in the system (when a trading operation is made between a buyer and a seller, we cannot assume that they will communicate their price to the rest of the bidders), prices are updated when a peer observes a bid or request from another peer. Clearly, the solution is not scalable and there are no guarantees that the information reaches all the peers. However, the mechanism can be useful for market implementations that do not require optimality and efficiency.

Esteva et.al. [Esteva and Padget(1999)] make use of a ring topology to distribute one side (English, Vickrey, and Dutch) auction processes amongst a set of brokers (called interagents). Interagents are mediators amongst buyers and the auctioneer and are responsible for receiving bids and clearing the auction. The clearing algorithm is based on the leader election algorithm [Lynch(1996)]. In short, the algorithm is based on finding the bidder with the highest bid, which will be the leader. The aim of introducing interagents is to reduce centralisation and the work of the auctioneer. However the authors do not indicate where interagents are executed and if they can be sellers or buyers participating in the auction. The authors point out that interagents have to be robust and introduce security measures against malicious peers. However, they do not introduce any security measure.

Atzmony and Peleg [Atzmony and Peleg(2000)] propose a set of algorithms for clearing English auctions in a distributed manner. They assume an underlying communication network represented by a complete n-vertex graph. Vertices represent the nodes in the network and every two vertices are connected with an edge that represents a bidirectional connection. Auctions are hosted by a subset of nodes in the graph. In the paper, they formally present a set of algorithms to clear an English auction in a distributed manner. The algorithm requires a static set of participants that join the auction before it starts. They also propose some enhancements to allow dynamic participants. Each auction is executed in several rounds until only one bidder remains. At the end of each round, new bidders are allowed to join the auction. Their asymptotic approach only finishes if no more bidders join the auction and all bidders except one resign. Although they formally verify the algorithm, the paper does not present any results on the performance and communication costs of the algorithm in a real network.

Several conclusions can be derived from those systems. Distribution of auction processes is costly. Decentralised auctions usually map one item to one responsible broker. In some other approaches, auction decentralisation can be compared to a way of sorting distribution problems, which can then be solved. The nature of auctions require complete information (i.e. the total set of bids is required) in order to determine the winner set and consequently decentralisation can only come when multiple items are traded assigning different responsible brokers for each item. In our work decentralisation is attained by short-lived market instances since we consider co-allocation and substitute allocation important aspects that need to be adressed.

7 Conclusions

In this paper a novel variant of the well-known Double Auction has been presented. The paper motivated the utility of the auction, as well as setting the context to be applied. MLDA constitutes an important contribution to the building of scalable and Cloud oriented service marketplaces. The paper presented the data structures used to design MLDA, as well as the main operations and algorithms that constitute the core of the auction. In order to evaluate the mechanism, an extensive set of experiments through simulation have been carried out. MLDA, due to its design, achieves optimal allocations in terms of Social Welfare.

The evaluation compared MLDA with another well-known implementation of the Double Auction. We tested MLDA and compared results with different configurations of the other mechanism. The results showed a better behavioural level for MLDA in all situations, due to the properties of the auction (i.e. invariant maximum social welfare). Moreover, MLDA has been compared with several double auction instances running at the same time. This experiment showed us the computational performance (i.e. time taken to execute) of MLDA when compared to a set of auctions potentially providing the same allocation. MLDA showed two to three times more computational efficiency that is MLDA is two to three times faster than multiple instances of single item double auctions.

Finally, MLDA showed to be a good candidate auction to trade time-differentiated services especially under the presence of substitute preferences. Besides MLDA can be considered a light-weight alternative to Combinatorial Auctions (CAs), being able to provide efficient allocations without dealing with the computational costs of CAs. The next step of the evaluation will compare MLDA with an instance of a Combinatorial Auction.

Acknowledgements. This work is being partially supported by the Grid4All European project under contract FP6-IST-034567 and Spanish MCYT project TIN2008-01288/TSI. This work has also been supported by the HAROSA Knowledge Community (http://dpcs.uoc.edu). Besides we feel deeply indebted with Leandro Navarro and Dirk Neumann for their helpful advice.

References

[Albrecht et al(2004)Albrecht, Patterson, and Vahdat] Albrecht, J., Patterson, D., Vahdat, A.: Distributed resource discovery on planetlab with sword. In: WORLDS - First Workshop on Real, Large Distributed Systems (2004)

[Atzmony and Peleg(2000)] Atzmony, Y., Peleg, D.: Distributed algorithms for english auctions. In: Herlihy, M.P. (ed.) DISC 2000. LNCS, vol. 1914, pp. 74–88. Springer, Heidelberg (2000)

[AuYoung et al(2004)AuYoung, Chun, Snoeren, and Vahdat] AuYoung, A., Chun, B., Snoeren, A., Vahdat, A.: Resource allocation in federated distributed computing infrastructures (2004), citeseer.ist.psu.edu/auyoung04resource.html

[Bao and Wurman(2003)] Bao, S., Wurman, P.R.: A comparison of two algorithms for multiunit k-double auctions. In: ICEC 2003: Proceedings of the 5th International Conference on Electronic Commerce, pp. 47–52. ACM, New York (2003), http://doi.acm.org/10.1145/948005.948012

[Buyya and Venugopal(2004)] Buyya, R., Venugopal, S.: The gridbus toolkit for service oriented grid and utility computing: An overview and status report (2004), citeseer.ist.psu.edu/buyya04gridbus.html

[Castro et al(2002)Castro, Druschel, Kermarrec, and Rowstron] Castro, M., Druschel, P., Kermarrec, A.M., Rowstron, A.: One ring to rule them all: service discovery and binding in structured peer-to-peer overlay networks. In: EW 2010: Proceedings of the 10th Workshop on ACM SIGOPS European Workshop: beyond the PC, pp. 140–145. ACM Press, New York (2002), http://doi.acm.org/10.1145/1133373.1133399

[Chun et al(2004)Chun, Ng, Albrecht, Parkes, and Vahdat] Chun, B.N., Ng, C., Albrecht, J., Parkes, D.C., Vahdat, A.: Computational resource exchanges for distributed resource allocation. Tech. rep. (2004)

[Chun et al(2006)Chun, Ng, Albrecht, Parkes, and Vahdat] Chun, B.N., Ng, C., Albrecht, J., Parkes, D.C., Vahdat, A.: Computational resource exchanges for distributed resource allocation (2006), citeseer.ist.psu.edu/706369.html

[Consortium(2008)] Consortium, G.: Grid4all european project (2008), http://grid4all.eu/

[Despotovic et al(2004)Despotovic, Usunier, and Aberer] Despotovic, Z., Usunier, J.C., Aberer, K.: Towards peer-to-peer double auctioning. In: HICSS 2004: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 9, p. 90289.1. IEEE Computer Society, Washington (2004)

[Esteva and Padget(1999)] Esteva, M., Padget, J.A.: Auctions without auctioneers: Distributed auction protocols. In: Agent Mediated Electronic Commerce (IJCAI Workshop), pp. 220–238 (1999), citeseer.ist.psu.edu/497541.html

[Eymann et al(2003)Eymann, Reinicke, Ardaiz, Artigas, Freitag, and Navarro] Eymann, T., Reinicke, M., Ardaiz, O., Artigas, P., Freitag, F., Navarro, L.: Decentralized resource allocation in application layer networks. ccgrid 00:645 (2003), http://doi.ieeecomputersociety.org/10.1109/CCGRID.2003.1199427

[Ghodsi(2006)] Ghodsi, A.: Distributed k-ary System: Algorithms for distributed hash tables. PhD dissertation, KTH—Royal Institute of Technology, Stockholm, Sweden (2006)

[Haussheer and Stiller(2005)] Haussheer, D., Stiller, B.: Decentralized auction-based pricing with peermart. In: Integrated Network Management, pp. 381–394. IEEE, Los Alamitos (2005)

[Kelly(1997)] Kelly, F.: Charging and rate control for elastic traffic (1997), citeseer.ist.psu.edu/kelly97charging.html

[Kevin Lai and Fine(2004)] Kevin Lai, B.A.H., Fine, L.: Tycoon: A Distributed Market-based Resource Allocation System. Tech. Rep. arXiv:cs.DC/0404013, HP Labs, Palo Alto, CA, USA (2004)

[Kwasnica et al(2005)Kwasnica, Ledyard, Porter, and DeMartini] Kwasnica, A.M., Ledyard, J.O., Porter, D., DeMartini, C.: A new and improved design for multiobject iterative auctions. Manage. Sci. 51(3), 419–434 (2005), http://dx.doi.org/10.1287/mnsc.1040.0334

[Lai et al(2005)Lai, Rasmusson, Adar, Zhang, and Huberman] Lai, K., Rasmusson, L., Adar, E., Zhang, L., Huberman, B.A.: Tycoon: An implementation of a distributed, market-based resource allocation system. Multiagent Grid Syst. 1(3), 169–182 (2005)

[Liu and He(2007)] Liu, Y., He, H.C.: Multi-unit combinatorial auction based grid resource co-allocation approach. In: International Conference on Semantics, Knowledge and Grid, vol. 0, pp. 290–293 (2007), http://doi.ieeecomputersociety.org/10.1109/SKG.2007.26

[Lynch(1996)] Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann Publishers Inc., San Francisco (1996)

[Mills and Dabrowski(2008)] Mills, K.L., Dabrowski, C.: Can economics-based resource allocation prove effective in a computation marketplace? Journal of Grid Computing 6, 291–311 (2008)

[Neumann et al(2007)Neumann, Stößer, Anandasivam, and Borissov] Neumann, D., Stößer, J., Anandasivam, A., Borissov, N.: Sorma - building an open grid market for grid resource allocation. In: Altmann, J., Veit, D. (eds.) GECON 2007. LNCS, vol. 4685, pp. 194–200. Springer, Heidelberg (2007)

[Nisan and Ronen(2001)] Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior 35, 166–196, 613 (2001)

[Phelps(2006)] Phelps, S.: Web site for JASA (Java Auction Simulator API) (2006), http://www.csc.liv.ac.uk/sphelps/jasa/

[Phelps(2007)] Phelps, S.: Evolutionary mechanism design. Ph. D thesis, University of Liverpool, U.K. (2007)

[Placek and Buyya(2006)] Placek, M., Buyya, R.: Storage exchange: A global trading platform for storage services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 425–436. Springer, Heidelberg (2006)

[Radhanikanth and Narahari(2009)] Radhanikanth, G.V.R., Narahari, Y.: Reverse combinatorial auction&#45;based protocols for resource selection in grids. Int. J. Grid Util. Comput. 1(2), 109–120 (2009), http://dx.doi.org/10.1504/IJGUC.2009.022027

[Ratnasamy et al(2001)Ratnasamy, Francis, Handley, Karp, and Schenker] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: SIGCOMM 2001: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 161–172. ACM, New York (2001), http://doi.acm.org/10.1145/383059.383072

[Schnizler(2007)] Schnizler, B.: Mace: A multi-attribute combinatorial exchange. In: Jennings, N., Kersten, G., Ockenfels, A., Weinhardt, C. (eds.) Negotiation and Market Engineering, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany. Dagstuhl Seminar Proceedings, vol. 06461 (2007), http://drops.dagstuhl.de/opus/volltexte/2007/1009 [date of citation: 2007-01-01]

[Schnizler and Neumann(2007)] Schnizler, B., Neumann, D.: Combinatorial exchanges for coordinating grid services. SIGecom Exch. 7(1), 65–68 (2007), http://doi.acm.org/10.1145/1345037.1345054

[Schwind et al(2006)Schwind, Gujo, and Stockheim] Schwind, M., Gujo, O., Stockheim, T.: Dynamic resource prices in a combinatorial grid system. In: CEC-EEE 2006: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, p. 49. IEEE Computer Society, Washington (2006), http://dx.doi.org/10.1109/CEC-EEE.2006.37

[Stoica et al(2001)Stoica, Morris, Karger, Kaashoek, and Balakrishnan] Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: Proceedings of the 2001 ACM SIGCOMM Conference, pp. 149–160 (2001), citeseer.ist.psu.edu/stoica01chord.html

[Tamai et al(2005)Tamai, Shibata, Yasumoto, and Ito] Tamai, M., Shibata, N., Yasumoto, K., Ito, M.: Distributed market broker architecture for resource aggregation in grid computing environments. In: CCGRID 2005: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), vol. 1, pp. 534–541. IEEE Computer Society, Washington (2005)

[Waldspurger et al(1992)Waldspurger, Hogg, Huberman, Kephart, and Stornetta] Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., Stornetta, S.: Spawn: A distributed computational economy. IEEE Transactions on Software Engineering 18(2), 103–117 (1992)

Decentralized Self-optimization in Shared Resource Pools

Emerson Loureiro[*], Paddy Nixon, and Simon Dobson

Abstract. Resource pools are collections of computational resources which can be shared by different applications. The goal with that is to accommodate the workload of each application, by splitting the total amount of resources in the pool among them. In this sense, utility functions have been pointed as the main tool for enabling self-optimizing behaviour in such pools. The goal with that is to allow resources from the pool to be split among applications, in a way that the best outcome is obtained. Whereas different solutions in this context exist, it has been found that none of them tackles the problem we deal with in a total decentralized way. In this paper, we then present a decentralized and self-optimizing approach for resource management in shared resource pools.

1 Introduction

Resource pools are collections of computational resources (e.g., servers) which can be used by different applications in a shared way[1]. The goal with that is to accommodate the workload of each application, by splitting the total amount of resources in the pool amongst them. This is possible through the use of Resource Containers[2], abstractions that can co-exist in a physical resource (e.g., server),

Emerson Loureiro · Paddy Nixon
Systems Research Group
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
e-mail: {emerson.loureiro,paddy.nixon}@ucd.ie

Simon Dobson
School of Computer Science
University of St Andrews,
St Andrews, United Kingdom
e-mail: sd@cs.st-sndrews.ac.uk

[*] This work is supported by UCD Ad Astra Scholarships.

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 149–170.
springerlink.com

each of them receiving a slice of it. Examples of resource containers include virtual machines and virtual disks. These resource containers are then aggregated into Resource Shares, thus forming partitions of the total amount of resources available in the pool.

In this scenario, the applications might have QoS parameters that have to be met. Therefore, the resources available to them should be such that their QoS parameters are met, if possible. The problem, in this case, is that the workload of the applications is likely to vary over time, and as a consequence, their resource demands will vary too[3][4]. Statically-defined resource shares, based for example on average or worst-case scenarios, are not suitable[5]. It is likely that resources will be wasted this way, for instance by allocating unnecessarily large shares and thus running the risk of failing to meet the applications' QoS. A better approach, instead, is to allow shares to be defined in an adaptive fashion, using the workload and QoS requirements of each application as input[1].

A usual trend, however, is not just to split the resources in the pool in a way that it meets the QoS parameters, but to do that in the best possible way. Precisely, that means finding the distribution of resources that yields the best outcome. To this end, utility functions have been pointed as the main tool for enabling such a self-optimizing behaviour[6], since they do not distinguish between desirable and undesirable allocations. Instead, allocations are distinguished by having a lower or higher utility, which then enables to find the best allocation, i.e., the one with the highest utility. Finding such an allocation consists, basically, on modelling the resource management process as an optimization problem, and eventually solving it. This has been called Utility Maximization (UM).

As a consequence of the above, employing Utility Maximization provides benefits, over other methods, when faced with conflicting scenarios. An example of such is during an overload in the system; i.e., the overall resource demand is greater than the amount of resources available to be allocated. In that case, it is clear that not all QoS requirements will be met. Still, with Utility Maximization, it is possible to find a way of maximizing resource usage, given the overload condition, thus providing directions as to how to act in such a conflicting scenario. In another case, the system could be facing a low load, in which case there might be several distributions of resources that meet all QoS requirements. Utility functions, again, provide unambiguous guidance towards the best way to do so. The important aspect, then, is that there will be situations where different resource distributions are possible. In these cases, unambiguous guidance as to how actually to do so is then crucial. Whereas other methods might be able to provide such guidance, Utility Maximization achieves that with an optimality aspect, regardless of the current setting.

A number of solutions employing Utility Maximization for managing shared resource pools have been proposed. Many are based on centralized architectures, which are known to be not very scalable and suffer from fault-tolerance issues, i.e., crash of the centralizer. Some distributed solutions have also been proposed. They are all modelled hierarchically though, and so, coordination is centralized at the root of the hierarchy. Given the increasing scale of distributed systems and a stronger demand in terms of their autonomy[7], a truly decentralized solution is preferable, since they provide improved scalability and are naturally fault tolerant. Whereas decentralized solutions in similar domains exist, they are not applicable to the problem being studied in this paper.

Given that, here we propose a truly Decentralized Utility Maximization (DUM) model for managing shared resource pools, in an adaptive and optimal way. To the best of our knowledge, this is the first work to present such a solution. For achieving that, we have employed the method of the Lagrange multipliers. Such methods have been used in similar works involving non-linear optimization. However, the problem being studied here along with the absolute decentralization characteristic of our DUM model, give it a crucial differential when compared to those works.

The rest of this paper is then organized as follows: in Section 2 some fundamental concepts are presented; our DUM model is presented in Section 3; an evaluation is presented in Section 4, demonstrating the feasibility of the model in a practical scenario, through simulations; related works in the area are presented and discussed in Section 5; finally, in Section 6, we conclude the paper with some final remarks and future directions of this work.

2 Fundamentals

In this section we provide basic concepts related to our DUM solution. More precisely, we present an overview of the ideas behind shared resource pools and also how utility functions are linked to these ideas, then leading to a proper formalization of the problem being studied here.

2.1 Shared Resource Pools and Related Paradigms

Shared resource pools are collections of computational resources, aggregated in a way to allow concurrent access to them[1]. This is done by splitting the resources from the pool into resource shares, or simply shares, which are in turn distributed to applications in need. In more practical terms, as discussed previously, each share is an aggregation of Resource Containers. Even though different terminologies can be found in the literature, e.g., Cluster Reserve[28], Server[29], and Application Environment[8], for the purposes of this paper, the term Resource Share will be used.

An example of a shared resource pool is illustrated in Figure 1. In this figure, a collection of servers, i.e., the pool, is split into shares – the solid lines surrounding the different sets of servers – which are then assigned to particular applications (APP in the figure). The same idea applies, for example, to a Distributed Rate Limiting scenario[20]. In this case, the pool is composed solely by the bandwidth capacity available. This capacity is then partitioned into shares to be provided to different traffic limiters. This way, they can serve their network flows in an optimal way, without, however, overusing a specific bandwidth capacity.

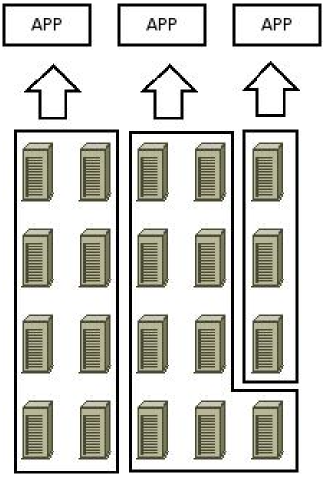


Fig. 1 Shared resource pool in a data center scenario

It is important to also point out that the idea of resource allocation and shared pools is found in the context of Utility Computing and Cloud Computing. Utility Computing is the on-demand packaging of computing resources so as to meet customers' needs, by dynamically creating virtual partitions of the resources available[30]. In this paradigm, the resource pool is viewed as a public infrastructure[31], in the sense that it is available to whoever has the need and is willing to pay for it.

The idea of Cloud Computing employs a similar model to go even further; moving away both data and computing from clients, placing them into large data centres[32], i.e., the cloud. The Cloud Computing paradigm, however, does not really focus on whether the infrastructure is public or private, in which case its resources are only available to the applications of whoever owns it[31]. Instead, its focus is more on providing a platform where not only hardware resources, but also applications, are provided as a service.

The work in this paper, putting it into the perspective of both paradigms, is neutral in terms of whether the infrastructure is public or private, like in Cloud Computing. Unlike Cloud, however, it is specifically focused on the delivery of hardware resources, and not applications. From the point of view of this work, applications can then be deployed as a service (e.g., a Google Doc-like spreadsheet application) or not (e-commerce application). Regardless of the access level to the resource infrastructure and nature of the applications, technically speaking, the allocation of resources can be done in the same way. In the end, it all boils down to consumers with resource demands over a common amount of resources. Consequently, the work presented here is clearly applicable to both the Utility and Cloud computing paradigms, even though it is being put in a more general context.

2.2 Problem Formalization

Firstly, because we are aiming at a decentralized approach, we view the system as a network of agents, where one agent can be reached by any other, directly or indirectly. In this case, each agent represents an application that consumes resources from the pool. We then denote by *S* the system itself and by a^i an agent in *S*, for $i \in [1, n(t)]$ where *n(t)* is the number of agents in the system at time *t*.

Secondly, for utility maximization purposes, our solution is based on the approach proposed in[8]. In this case, each agent a^i is assigned a utility function $u^i(x)$, stating how useful a resource share *x* from the pool is at a particular point in time. From that, a collective utility function *U(X)* is defined, as follows:

$$U(X) = \sum_{a_i \in S} u^i(X_i), \tag{1}$$

where $X = \{X_1, X_2, \ldots, X_{n(t)}\}$ is an allocation vector and X_i is the resource share assigned to agent a^i. In practical terms, X_i could be, for example, the number of servers or amount of bandwidth allocated to a particular agent. Such an approach then maps every possible distribution of resources to a real-scalar value, which is used to distinguish between two different allocations. To find the best allocation at any point in time, the following optimization model, proposed in[9][10], is used:

$$\begin{aligned} &\max_{X \in R^{n(t)}} U(X) \\ &\text{subject to}: \sum_{\mathrm{i}=1}^{|\mathrm{X}|} \mathrm{X}_{\mathrm{i}} = K(t) \end{aligned}, \tag{2}$$

where *K(t)* is the amount of resources available in the pool at time *t*, e.g., 100 servers. The constraint limits the sum of all resource shares to *K(t)*. In a practical setting, the value of *K(t)* could be set by system administrators from a management station, then being propagated throughout the system[11].

3 DUM Model

In this section we present our solution for decentralized self-optimization in shared resource pools. More precisely, our solution consists on how to solve the optimization problem in Equation 2 in a truly decentralized way. For that, first, the utility function of the agents is defined, as follows:

$$u^i(x) = 1 - e^{-\alpha^i(t)x}, \tag{3}$$

where *x* is the amount of resource from the pool being allocated to a^i and $\alpha^i(t)$ is a parameter that indicates a^i's resource demand at time *t*. The smaller $\alpha^i(t)$ is, the greater is the agent's resource demand. The reason for using such a utility function is because it will enable us to break down the optimization problem into separate models that each agent can use to find its optimal share. Like ours, other works have also used specific utility functions for different purposes[9][12].

Some plots of $u^{i}(x)$ are presented in Figure 2. The sharpness of the utility is controlled by $\alpha^{i}(t)$. The less sharp the utility is, the smaller is $\alpha^{i}(t)$, thus indicating a greater demand for resources. We assume $\alpha^{i}(t)$ might, and most likely will, change over time. However, it should remain constant during the actual process of finding the optimal allocation, i.e., solving the problem in Equation 2. Since $\alpha^{i}(t)$ represents an agent's resource demand at that particular time slot, it does not make much sense for it to change within such an interval. This is a similar requirement for the centralized case, where a central entity, a Solver in this case, solves the optimization problem. In this case, once the Solver starts trying to find the solution for the optimization problem, the variables are not allowed to change until it is finished, or the solution found will just not be the correct one. That is then not a limiting factor from our DUM model, but simply something inherent to the scenario we are dealing with.

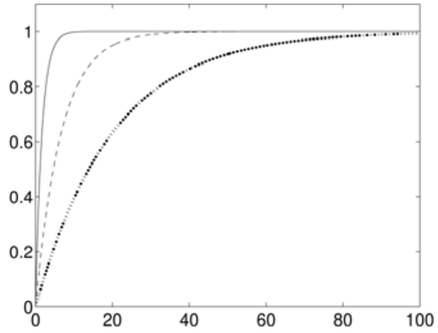


Fig. 2 Sample plots of the utility of the agents

From $u^{i}(x)$, we then transform the constrained optimization problem in Equation 2 into an unconstrained one. Using the method of the Lagrange multipliers, the new problem can be formulated as:

$$\max_{X \in R^{n(t)}, \lambda \in R} L(X, \lambda), \tag{4}$$

where $L(X, \lambda)$ is the Lagrangian of the problem in Equation 2, being defined as:

$$L(X, \lambda) = U(X) - \lambda\left(\left(\sum_{i=1}^{|X|} X_i\right) - K(t)\right). \tag{5}$$

We then solve 4, in a way that it decomposes into the models that will calculate each agent's optimal share. For that, we can solve

$$\nabla L(X,\lambda)=0\,,$$

which gives us the set of equations below.

$$\begin{gathered}\frac{\partial L}{\partial X_i}=0,\forall i\in[1,|X|]\\ \frac{\partial L}{\partial\lambda}=0.\end{gathered}\tag{6}$$

In theory, though, two points must be highlighted. First, for *X* to be an optimal solution of the optimization problem, it must satisfy the Karush-Kuhn-Tucker (KKT) conditions. Second, by solving $\nabla L(X, \lambda) = 0$, we would actually find a set of stationary points, each of which being a maximum, a minimum, or a saddle point. It can be shown, however, that in our case, the solution to $\nabla L(X, \lambda) = 0$ satisfies the KKT conditions, is unique, i.e., only one stationary point exists, and also that such a stationary point is necessarily a maximum, and consequently the global maximum. The proofs for those can be found in the Appendix. Back to Equations 6, each $\partial L/\partial X_i = 0$ will yield in:

$$\alpha^i(t)e^{(-\alpha^i(t)X_i)}-\lambda=0\,.$$

Solving the equation above for X_i, gives us:

$$X_i=\frac{\ln\alpha^i(t)-\ln\lambda}{\alpha^i(t)}\,,\tag{7}$$

which then enables each agent to find its own share, such that *U(X)* in Equation 2 is maximized. Note, first, that $X_i \in R$, and so, fine-grained shares are supported. Second, coordination in this case is totally decentralized. To calculate X_i, however, agents need, besides their own $\alpha^i(t)$, the value of $\ln\lambda$, which is the global information that binds them together. Therefore, to compute their shares, they would need to compute $\ln\lambda$ first, also in a decentralized way. For that, we start with $\partial L/\partial\lambda = 0$, from Equation 6, which yields in:

$$\left(\sum_{i=1}^{|X|}-X_i\right)+K(t)=0\,.$$

Substituting 7 in the above, we then have that:

$$\left(\sum_{i=1}^{|X|}\frac{\ln\lambda-\ln\alpha^i(t)}{\alpha^i(t)}\right)+K(t)=0\,.$$

We can isolate *ln* λ, ending up with:

$$\ln \lambda = \frac{\left(\sum_{i=1}^{|X|} \frac{\ln \alpha^i(t)}{\alpha^i(t)} \right) - K(t)}{\sum_{i=1}^{|X|} \frac{1}{\alpha^i(t)}} . \tag{8}$$

With that, each agent can then calculate *ln* λ, and, once that is done, their own share through Equation 7.

Because *ln* λ depends on the α of all agents, and because we do not want any kind of centralization in the system, we assume that either each $\alpha^i(t)$ will be disseminated throughout the system, eventually reaching every other agent[13], or *ln* λ will be computed using approaches for calculating aggregates in networked systems[14][15].

In the first case, each agent will end up with the α of the others, which are then combined with its own and used as input to Equation 8. In the second one, each agent a^i would hold two values, $ln\ \alpha^i(t)/\alpha^i(t)$ and $1/\alpha^i(t)$. From that, one run of an aggregate algorithm would be executed for each value, to perform a sum of all of such values. When the two sums are computed, each agent uses them appropriately in Equation 8, so as to find their own share. Both approaches can be performed in large-scale networks in very reasonable time, thus not compromising our solution in terms of performance. Further discussion on the actual algorithms for computing *ln* λ, however, is out of the scope of this paper.

4 Evaluation

In this section we present experiments we have performed using our DUM model. To this end, we have modelled a scenario where a number of Application Environments (AEs) are deployed in a data center, as proposed in[9], each AE processing one type of transaction. The scenario we illustrate here will then deal with the allocation of servers from the data center to the AEs deployed in it. In this case, each AE is represented by an agent implementing our DUM model.

4.1 Data Center Model

Each AE has an Expected Average Workload (EAW) at different points in time, in terms of number of requests per second. That can be obtained using online or offline prediction techniques. For our experiments, these workloads have been obtained from the analytical data of different web sites. Also, all AEs have a policy defining a Target Response Time (TRT) that should be guaranteed for the transactions they process. The resource management process will then find the optimal distribution of servers amongst the AEs, considering their EAW and TRT.

We denote by $r^i(s,w)$ the Expected Average Response Time (EART) of an AE during time *t*, representing the response time an AE will obtain given a workload *w* and a certain number of servers *s* allocated to it. We define $r^i(s,w)$ based on the model proposed in[9], as:

$$r^i(s,w) = \frac{w\,c^i}{s}, \tag{9}$$

where c^i is the CPU time of the transaction processed by AE *i* (in seconds), *w* is the EAW of the AE (in requests per second), and *s* is the amount of servers assigned to it. From that, we derive $q^i(w)$, the required amount of servers that should be assigned to an AE, in order to meet its TRT, as follows:

$$q^i(w) = \frac{w\,c^i}{T^i}, \tag{10}$$

where *w* and c^i are as in $r^i(s,w)$ and T^i is the AE's TRT. The required amount of servers $q^i(w)$ is necessary for defining $\alpha^i(t)$, which, according to our DUM model, represents an AE's resource demand at a particular time *t*. Such a parameter is calculated as:

$$\alpha^i(t) = -\frac{\ln(1-H)}{q^i(w)}, \tag{11}$$

where H represents the value of the agents' utility when the EART of its AE meets its TRT, i.e., a value very close to 1. This then models the fact that agents are "happy" with an amount of resources that causes their TRT to be met, but also being "happier" if more resources are given.

4.2 Simulation Results

Based on the data center model presented, a series of experiments have been run, using different scenarios. In these experiments, a random epidemic algorithm for disseminating all $\alpha^i(t)$ has been used. Also, we assumed that the resource management process runs at distinct points in time, called iterations. In a real world setting, these iterations could represent different hours of the day, on which a reallocation of the servers would take place. The results for the experiments are then presented next.

4.2.1 Scenario 1: Static Number of AEs

In this scenario, the number of AEs over the entire simulation is constant. We considered that six AEs, whose EAWs are as in Figure 3, are deployed in the data center. Also, we assumed that 145 servers are available on the data center and that the CPU times of the transactions processed by each AE are as presented in Table 1. The latter has been based on values provided in[9].

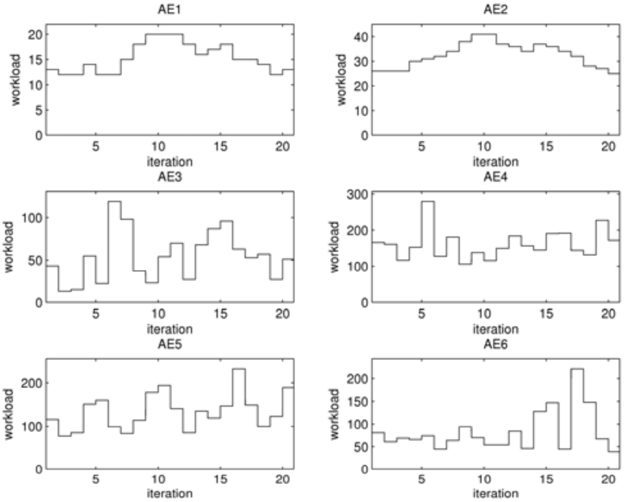


Fig. 3 EAWs of each AE in Scenario 1

Table 1 CPU Times (in seconds) for the transactions processed by the AEs

AE	CPU Time
1	0.11
2	0.015
3	0.045
4	0.08
5	0.01
6	0.096

After running the simulation during twenty iterations, the shares found by each agent were as presented in Figure 4. The important aspect to note is the way the shares vary. Note that, the general shape of the graphs of the shares vary similarly to the way the workload does. Therefore, from a high-level perspective, our DUM solution captures the demands correctly, and acts properly towards the optimal share. At a lower level, one can see that, sometimes variations between the shares and workload do not match. In a general way, it is clear that those were the variations that yielded in the highest *U(X)*, even though the specific reasons for such can vary. As an example, notice that, at iteration 5, AEs 2, 4, 5, and 6 have an increase on their workload, but only AE 5 has an increase on its share. That is because the workload increase in AE 5 was simply too high to allow an increase in the shares of AEs 2, 4, and 6 such that *U(X)* would be maximized.

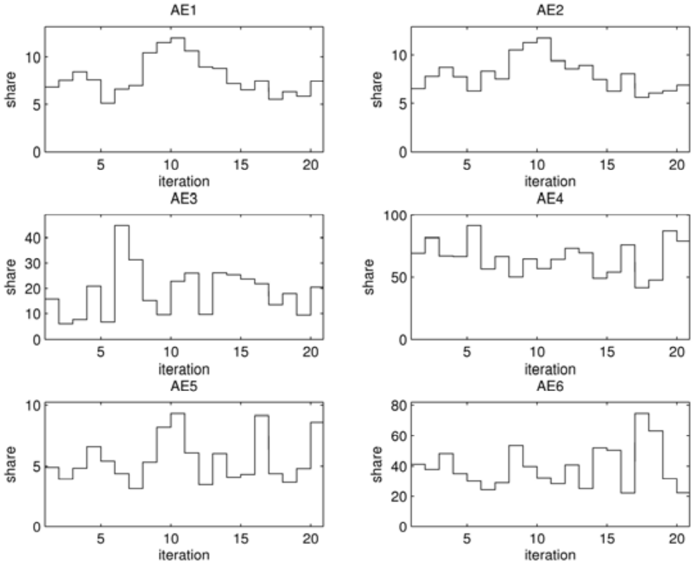


Fig. 4 Shares of each AE in Scenario 1

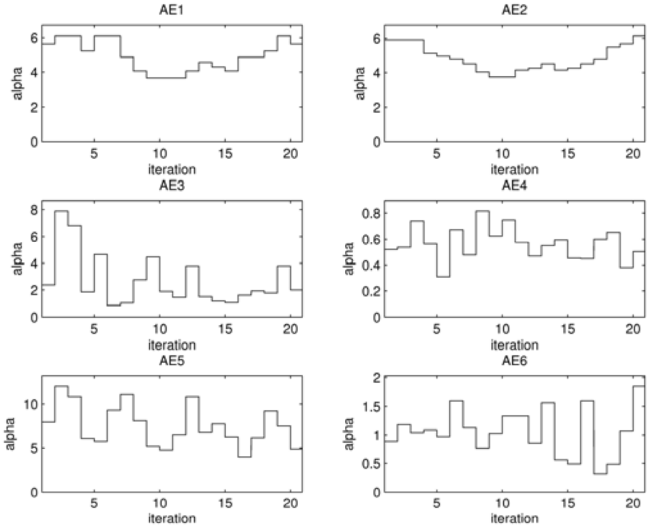


Fig. 5 $\alpha^i(t)$ of each AE in Scenario 1

The fact that our DUM model captures demand correctly is reinforced by the results presented in Figure 5, where the $a^i(t)$ of each AE over all iterations are presented. Note that the values of $a^i(t)$ vary exactly the opposite to the way the workload does. This thus matches the definition of the agents' utility function, on which it is stated that the greater the workload is, the smaller is the value of $a^i(t)$, indicating a greater demand for resources.

As a consequence of properly capturing $a^i(t)$, the EARTs for all AEs end up as in Figure 6. Note that, for all AEs, such response times are always smaller than what is specified in their TRT, represented by the dashed horizontal line in each graph of the figure. Because, in this scenario, the data center always hosted more servers than the demand, the aggregate utility was always such that $U(X) \approx 6$, i.e., the maximum under any condition.

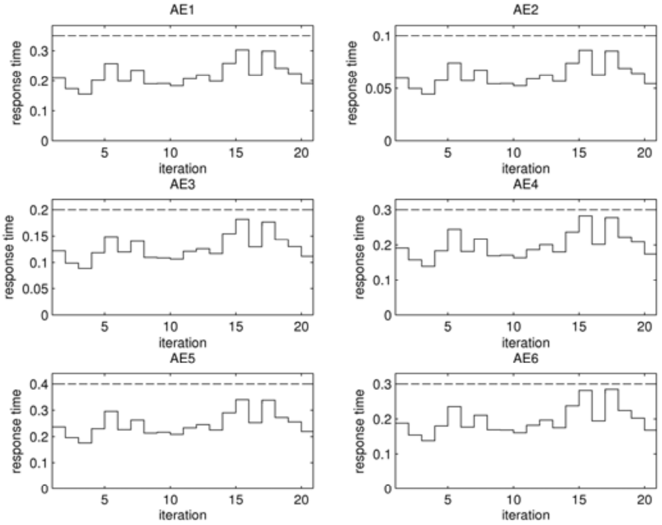


Fig. 6 EARTs of each AE in Scenario 1

4.2.2 Scenario 2: Varying Number of AEs

In a real world setting, we cannot expect the system to be static. As it evolves, AEs will join and leave the data center. Consequently, our solution should support such dynamics, which is what we have simulated in this scenario. To this end, the data center was initially set up with four AEs, until iteration ten, when two AEs join the system. Then, at iteration fifteen, one of them leaves the system, keeping this setting until the end of the simulation. The number of servers and the CPU times for this scenario are the same as for the first one.

The EAWs for this scenario are then illustrated in Figure 7. After running the simulation, the shares found were such that the EARTs in Figure 8 were obtained. As with the previous scenario, note that the EARTs of each AE is always smaller than their TRT (dashed horizontal line in each graph). This demonstrates that our DUM model supports these changes smoothly. The figures for the $\alpha^i(t)$'s and shares found were similar to the ones presented in the first scenario, and so we omitted them.

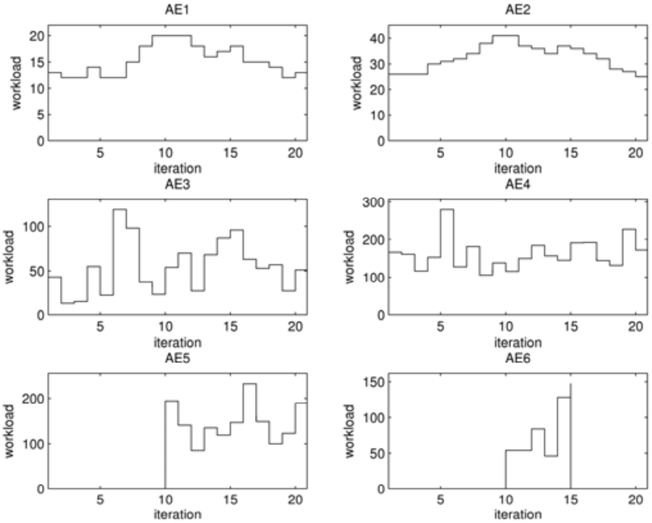


Fig. 7 EAWs of each AE in Scenario 2

4.2.3 Scenario 3: Varying QoS Parameters

Just like the system structure is prone to changes in real world settings, so are the QoS requirements of the applications in a shared resource pool. Contracts between customers and infrastructure providers might change over time, thus directly impacting in resource demands. We should then be able to show that our DUM model can also handle such variations in QoS requirements, so as to make it fully applicable in practice.

For that, we have run simulations similar to the ones for the first scenarios, but this time varying the TRT of the AEs deployed in the data center. The workloads of the AEs, as well as the number of servers and the CPU times of the transactions, were the same as for the first scenario. The variation of the TRTs, along with the final EARTs obtained, for each AE, is presented in Figure 9. Note that, even though the TRTs were varying over time, our DUM model was still able to deliver resource shares meeting all such requirements, for all AEs. It is worthy to point out, in this case, that since not only the workload, but also the QoS requirements, are varying, the $\alpha^i(t)$ will now vary based on a combination of both, as presented in Figure 10. This contrasts with the first scenario, and it does so because in that case the TRTs were constant over time, and so, only the workload would affect the $\alpha^i(t)$. In that case then, it would be easy to realize the behaviour of $\alpha^i(t)$, i.e., an increase/decrease in the workload, would necessarily cause an increase/decrease in resource demand, ending up with a decrease/increase of $a^i(t)$. For this scenario, we omitted the figures for the shares obtained, since they present results similar to what has already been shown.

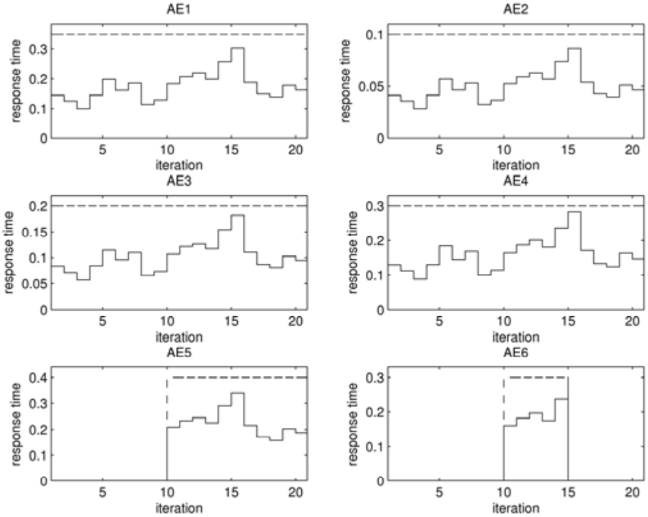


Fig. 8 EARTs of each AE in Scenario 2

4.2.4 Scenario 4: Overload

Finally, in a fourth scenario, we observed how our solution behaves when facing overload in the data center. In other words, in some iterations, we allowed the total demand to be greater than the number of servers available in the Data Center. For that, the number of servers has been set to 100. Again, the CPU times used and the workloads were as in the first scenario.

The overload is illustrated in Figure 11, which plots the variation of the total server demand over the iterations (the solid line represents the number of servers available). Because of that, the EARTs were then as in Figure 12. Since overloading was being considered, in some iterations, the TRTs of some, or all, AEs could not be met. To better illustrate that, we present in Figure 13 the variation of the aggregated utility. Note that, on the iterations where overload did not happen, the utility obtained was still the highest possible, i.e., $U(X) \approx 6$, consequently decreasing during overload periods. Consequently, the point where the aggregate utility reached its lowest value was the exact moment where the total server demand reached its highest value.

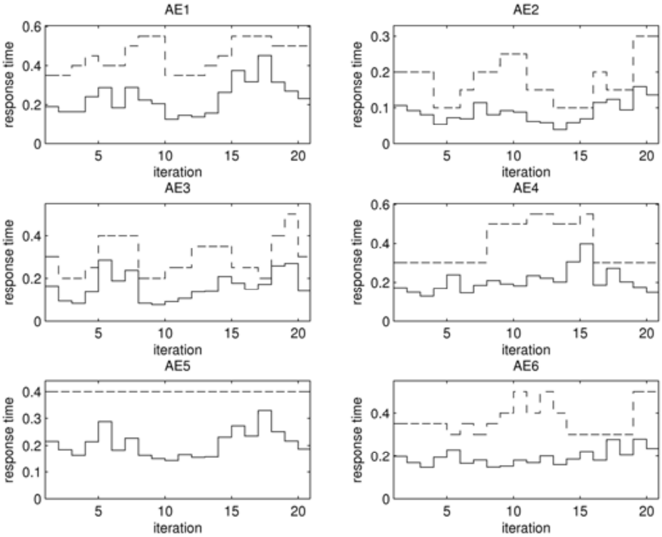


Fig. 9 TRTs and EARTs of each AE in Scenario 3

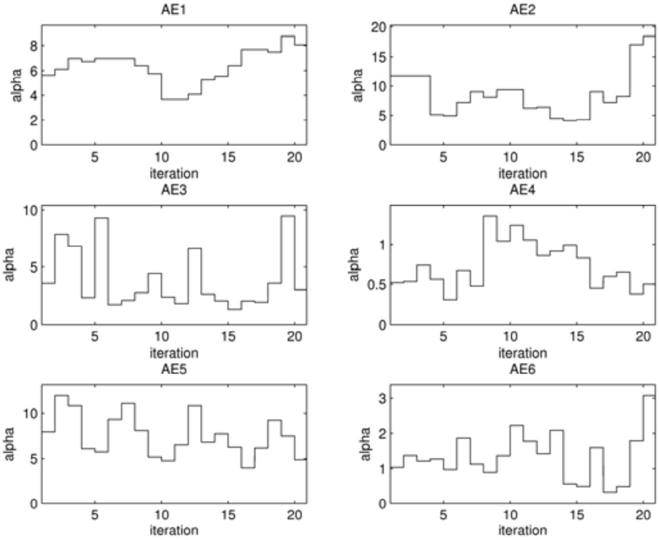


Fig. 10 $a^i(t)$ of each AE in Scenario 3

Still, our DUM model distributed the shares in a way that always maximized the aggregated utility, as stated in the original problem formulation. Apart from the formal proofs provided in the Appendix concerning the maximization of the problem formulation, we provide, in Table 2, practical evidence that the allocations found by our DUM solution do maximize the aggregated utility. More precisely, we compared the optimal allocation found, lets call it X^*, for a particular run, with different allocations, lets call them $X^{'}$, by giving/taking 0.1 resource shares to/from the AEs, in a way that the total amount of resources allocated did not change. Then we compared the value of the aggregated utility *U(X)* for the two allocations. As one can see in Table 2, the value obtained with the allocation X^* is always higher. That then further demonstrates that such an allocation, found by our DUM model, is, indeed, the one yielding to the maximum of the original optimization problem.

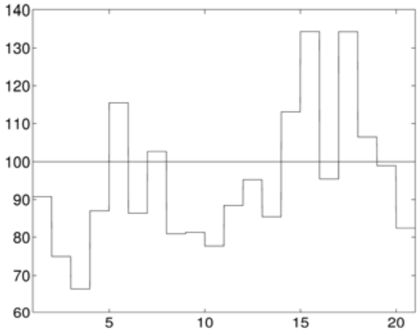


Fig. 11 Total amount of required servers over the overload scenario

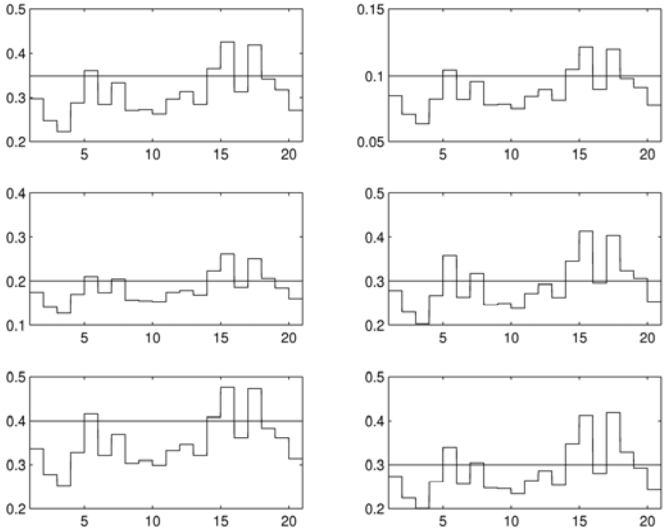


Fig. 12 EART of the AEs over the overload scenario

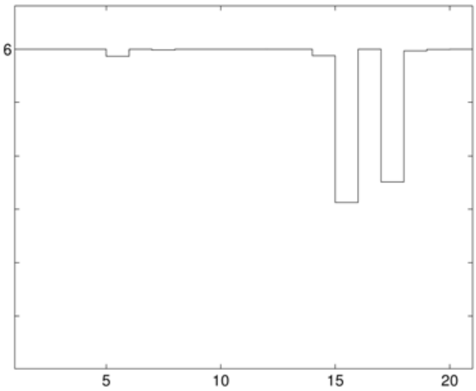


Fig. 13 Aggregate utility over the overload scenario

Table 2 Comparison of the values of the aggregate utility *U(X)*

U(X) - U(X')*
6.767E-11
5.678E-11
1.191E-10
1.318E-10

5 Related Work

A number of solutions for performing self-optimizing resource management in shared resource pools has been proposed. Many of them, however, approach the problem using centralized models[3][9][5]. In this case, a central authority is in charge of deciding the resource shares across the system. Even though these solutions can perform well, they are not very scalable because of the centralizers. Also, they are not fault-tolerant, since the crash of the centralizer compromises the entire system.

Solutions with a more distributed characteristic have also been proposed. In[12], for example, market agents are used. The solution differs from ours in the sense that centralizing entities, called brokers, are inserted in the resource management process. The decomposition methods presented in[16] are another distributed solution. These methods are similar to our DUM model in that they also employ Lagrange multipliers to decompose an optimization problem into smaller problems, which can further decomposed, forming a hierarchy. Unlike our solution, it relies on a messaging scheme which employs a central problem. Similarly, a hierarchical optimization model is presented in[17]. In both cases, coordination is done at the root of the hierarchy, whereas in our case, this is decentralized.

Solutions featuring decentralized control exist in similar domains. Examples of such solutions are [18][19], which employ market agents. Their focus, however, is not on Utility Maximization, unlike our DUM model. In [11], it is presented a decentralized solution for allocating servers to different classes of service. This solution is modelled differently though, in that resource providers, and not consumers, solve the optimization problem, like in our DUM model. Besides, it is specifically focused on server allocation, whereas we have aimed at a more general approach. In [20], gossiping is used to allow a set of P2P-connected traffic limiters to control the bandwidth they use. The solution is different from ours in the sense that it does not focus on Utility Maximization. The same can be said from the approach proposed in [21], which allows servers to be allocated to applications in a decentralized way. In terms of distributed optimization, in [22], Subgradient methods are used to optimize the aggregate of a set of agents' cost function. The solution, however, does not incorporate resource constraints, limiting its applicability in practical resource management scenarios. Finally, in [23], a DUM model is proposed, but it is focused on the control of multiple multicasts in P2P systems and does not apply to the problem formulation being used here.

6 Conclusions

In this paper, we presented a Decentralized Utility Maximization (DUM) model for managing shared resource pools in an adaptive and optimal way. More precisely, we employed the method of the Lagrange multipliers along with the utility functions theory to devise a method where each agent in the system knows how to calculate its share, so that the best outcome can be obtained. As we showed, centralized and hierarchical solutions exist in this context, but none of the decentralized ones cope with the specific problem being studied here. That thus gives our DUM model an innovative feature. To the best of our knowledge, this is the first work to present a decentralized solution in the domain of shared resource pools.

An evaluation has been presented, through simulations, using a server allocation scenario in a data center. We demonstrated that our DUM model is able to capture resource demands properly and deliver shares that meet all applications' QoS parameters, when possible. Scenarios where the number of applications as well as the QoS parameters in the system vary, which are to happen in the real world, have been simulated. As we showed, our DUM model also handles these scenarios in an optimal way. Finally, in overload situations, even though not all QoS parameters could be met, we demonstrated that our solution was still able to find the allocation leading to the best outcome.

As future work, we are aiming at a specific epidemic algorithm for disseminating the $\alpha^i(t)$ values throughout the system. The main reason for such is that the current methods for computing aggregates like our $ln\ \lambda$ would not suit us in terms of scalability, precision, and fault-tolerance. Furthermore, we will apply our DUM model to other shared resource pools scenarios, to have an insight on how general it really is. We do believe, however, that our DUM model could handle other scenarios straightforwardly.

References

1. Rolia, J., Cherkasova, L., Arlitt, M., Machiraju, V.: Supporting application quality of service in shared resource pools. Communications of the ACM 49(3), 55–60 (2006)
2. Banga, G., Druschel, P., Mogul, J.C.: Resource containers: A new facility for resource management in server systems. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation. USENIX Association, pp. 45–58 (1999)
3. Wang, X., Du, Z., Chen, Y., Li, S.: Virtualization-based autonomic resource management for multi-tier web applications in shared data center. Journal of Systems and Software 81(9), 1591–1608 (2008)
4. Guitart, J., Carrera, D., Beltran, V., Torres, J., Ayguade, E.: Dynamic CPU provisioning for self-managed secure web applications in smp hosting platforms. Computer Networks 52(7), 1390–1409 (2008)
5. Padala, P., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., Salem, K.: Adaptive control of virtualized resources in utility computing environments. In: Proceedings of the 2007 European Conference on Computer Systems, pp. 289–302. ACM Press, New York (2007)
6. Kephart, J.O., Das, R.: Achieving self-management via utility functions. IEEE Internet Computing 11(1), 40–48 (2007)
7. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36(1), 41–50 (2003)
8. Tesauro, G., Kephart, J.O.: Utility functions in autonomic systems. In: Proceedings of the First International Conference on Autonomic Computing, pp. 70–77. IEEE Computer Society, Washington (2004)
9. Bennani, M.N., Menasce, D.A.: Resource allocation for autonomic data centers using analytic performance models. In: Proceedings of the Second International Conference on Autonomic Computing, pp. 229–240. IEEE Computer Society, Washington (2005)
10. Tesauro, G., Walsh, W.E., Kephart, J.O.: Utility-function-driven resource allocation in autonomic systems. In: Proceedings of the Second International Conference on Autonomic Computing, pp. 342–343. IEEE Computer Society, Washington (2005)
11. Johansson, B., Adam, C., Johansson, M., Stadler, R.: Distributed resource allocation strategies for achieving quality of service in server clusters. In: Proceedings of the 45th Conference on Decision and Control, pp. 1990–1995. IEEE Computer Society, Washington (2006)
12. Bai, X., Marinescu, D.C., Boloni, L., Siegel, H.J., Daley, R.A., Wang, I.J.: A macroeconomic model for resource allocation in large-scale distributed systems. Journal of Parallel Distributed Computing 68(2), 182–199 (2008)
13. Kermarrec, A.M., Van Steen, M.: Gossiping in distributed systems. SIGOPS Oper. Syst. Rev. 41(5), 2–7 (2007)
14. Jelasity, M., Kowalczyk, W., van Steen, M.: An approach to massively distributed aggregate computing on peer-to-peer networks. In: Proceedings 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, pp. 200–207. IEEE Computer Society, Washington (2004)
15. Kempe, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science. IEEE Computer Society, Washington (2003)

16. Palomar, D.P., Chiang, M.: A tutorial on decomposition methods for network utility maximization. IEEE Journal on Selected Areas in Communications 24(8), 1439–1451 (2006)
17. Nowicki, T., Squillante, M.S., Wu, C.W.: Fundamentals of dynamic decentralized optimization in autonomic computing systems. In: Babaoğlu, Ö., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., van Moorsel, A., van Steen, M. (eds.) SELF-STAR 2004. LNCS, vol. 3460, pp. 204–218. Springer, Heidelberg (2005)
18. Lewis, P.R., Marrow, P., Yao, X.: Evolutionary market agents for resource allocation in decentralised systems. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1071–1080. Springer, Heidelberg (2008)
19. Maheswaran, R., Basar, T.: Nash equilibrium and decentralized negotiation in auctioning divisible resources. Group Decision and Negotiation 12(5), 361–395 (2003)
20. Raghavan, B., Vishwanath, K., Ramabhadran, S., Yocum, K., Snoeren, A.C.: Cloud control with distributed rate limiting. SIGCOMM Comput. Commun. Rev. 37(4), 337–348 (2007)
21. Masuishi, T., Kuriyama, H., Ooki, Y., Mori, K.: Autonomous decentralized resource allocation for tracking dynamic load change. In: Proceedings of the 7th International Symposium on Autonomous Decentralized Systems, pp. 277–283 (2005)
22. Nedic, A., Ozdaglar, A.: Distributed Subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control 54(1), 48–61 (2009)
23. Chen, M., Ponec, M., Sengupta, S., Li, J., Chou, P.A.: Utility maximization in peer-to-peer systems. In: Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 169–180. ACM, New York (2008)
24. Kuhn, M.: The Karush-Kuhn-Tucker theorem, `http://webrum.uni-mannheim.de/vwl/mokuhn/public/KarushKuhnTucker.pdf`
25. Weisstein, E.W.: Concave function, `http://mathworld.wolfram.com/ConcaveFunction.html`
26. Weisstein, E.W.: Convex function, `http://mathworld.wolfram.com/ConvexFunction.html`
27. Gallini, A.: Affine function, `http://mathworld.wolfram.com/A_neFunction.html`
28. Aron, M., Druschel, P., Zwaenepoel, W.: Cluster reserves: a mechanism for resource management in cluster-based network servers. In: Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 90–101. ACM Press, New York (2000)
29. Loureiro, E., Nixon, P., Dobson, S.: A fine-grained model for adaptive on-demand provisioning of CPU shares in data centers. In: Hummel, K.A., Sterbenz, J.P.G. (eds.) IWSOS 2008. LNCS, vol. 5343, pp. 57–108. Springer, Heidelberg (2008)
30. Lysne, O., Reinemo, S.A., Skeie, T., Solheim, A.G., Sodring, T., Huse, L.P., Johnsen, B.D.: Interconnection networks: Architectural chalenges for utility computing data centers. Computer 41(9), 62–69 (2008)
31. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A Berkeley view of cloud computing. Technical report, University of California at Berkeley (2009)
32. Dikaiakos, M.D., Katsaros, D., Mehra, P., Pallis, G., Vakali, A.: Cloud computing: Distributed internet computing for it and scientific research. IEEE Internet Computing 13, 10–13 (2009)

Appendix

A. Satisfiability to the Karush-Kuhn-Tucker (KKT) Conditions

Consider the following non-linear optimization problem:

$$\max_{x \in R^n} f(x) \\ \text{subject to}: g_i(x) \leq 0, h_j(x) = 0 ,$$

where $g_i(x)$, $i \in [1, m]$, are inequality constraints and $h_j(x) = 0$, $j \in [1, n]$, are equality constraints. It is known that if $f(x)$ and all $g_i(x)$ are concave and all $h_j(x)$ are affine functions, then, $\nabla L(X, \lambda) = 0$ is a sufficient condition for a maximum[24], i.e., it satisfies the KKT conditions. Translating that into our optimization problem, then:

$$f(x) = U(X) \\ h(x) = \sum_{a^i \in S(t)} X_i - K(t) ,$$

since we have no inequality constraints. We then start by showing that $U(X)$ is concave. For that, we can simply show that $-U(X)$ is strictly convex [25]. If a function $f(x)$ has a second derivative, for it to be strictly convex it is necessary and sufficient that $\forall x, f''(x) \geq 0$ [26]. We know that:

$$\nabla U(X) = \left(\frac{\partial U}{\partial X_1}, \ldots, \frac{\partial U}{\partial X_{n(t)}} \right) \\ \nabla - U(X) = -\left(\alpha^1(t) e^{-(\alpha^1(t) X_1)}, \ldots, \alpha^{n(t)}(t) e^{-(\alpha^{n(t)}(t) X_{n(t)})} \right) .$$

Consequently

$$\nabla^2 - U(X) = -\left(-\alpha^1(t)^2 e^{-(\alpha^1(t) X_1)}, \ldots, -\alpha^{n(t)}(t)^2 e^{-(\alpha^{n(t)}(t) X_{n(t)})} \right) \\ \nabla^2 - U(X) = \left(\alpha^1(t)^2 e^{-(\alpha^1(t) X_1)}, \ldots, \alpha^{n(t)}(t)^2 e^{-(\alpha^{n(t)}(t) X_{n(t)})} \right) , \quad (21) \\ \nabla^2 - U(X) \geq 0$$

which is what we wanted to show. Finally, showing that h(x) is affine is straightforward. Any affine function is of the following form [27]:

$$f(x_1, \ldots, x_n) = A_1 X_1 + \ldots + A_n X_n + b ,$$

where A_i can be a scalar. Clearly, then, $h(x)$ is affine, since

$$h(x_1, \ldots, x_n) = X_1 + \ldots + X_n + (-K(t)) ,$$

where $A_1 = A_2 = \ldots = A_n = 1$. By showing that, then, we have that $\nabla L(X, \lambda) = 0$ is a sufficient condition for a maximum.

B. Existence and Uniqueness of the Maximum of *L(X, λ)*

Based on the optimization problem defined in Equation 4, we now show that, by solving $\nabla L(X, \lambda) = 0$ in the Optimization Model, we are not only finding a maximum of $L(X, \lambda)$, instead of a minimum or a saddle point, but also that such a maximum is unique, which then makes it the global maximum. For that, we start showing that $L(X, \lambda)$ is strictly concave. From the general non-linear optimization problem given in Appendix A, $L(X, \lambda)$ is strictly concave if $f(x)$, $g_i(x)$, and $h_j(x)$ are strictly concave[24]. From Equations 21, we already have that $f(x) = U(X)$ is strictly concave. Because we have no $g_i(x)$ constraints, it only remains for us to show that $h(x) = \Sigma_{ai \in S(t)} X_i$ - $K(t)$ is strictly concave. For that, as in Appendix A, we have to show that $-h(x)$ is strictly convex, which can be done by checking if $\forall x$, $-h''(x) \geq 0$, if $-h''(x)$ is defined. We then have the following:

$$\nabla - h(x) = -(-K(t)_1, ..., K(t)_{n(t)})$$
$$\nabla^2 - h(x) = (0, ..., 0) \quad ,$$

which makes $h(x)$ strictly concave. Now that we know $L(X, \lambda)$ is indeed strictly concave, the following can be stated:

1. A stationary point of $L(X, \lambda)$, i.e., X', λ' such that $L(X', \lambda') = 0$, if any, is necessarily a maximum;
2. For a X', λ' such that $L(X', \lambda') = 0$, it can be said that X', λ' are unique,

which then ensures what we wanted to show.

Monitoring and Performance Analysis of Workflow Applications in Large Scale Distributed Systems

Dragos Sbirlea, Alina Simion, Florin Pop, and Valentin Cristea

Abstract. The chapter presents the design, implementation and testing of the monitoring solution created for integration with a workflow execution platform. The monitoring solution is a key for modeling and performance analysis of Grid systems considered as a networking and collaborative systems. The monitoring solution constantly checks the system evolution in order to facilitate performance tuning and improvement. The novelty of the work presented in this chapter is the improvement of distributed application obtained using the real-time information to compute estimates of runtime which are used to improve scheduling. Monitoring is accomplished at application level, by monitoring each job from each workflow and at system level, by aggregating state information from each processing node. The scheduling performance in distributed systems can be improved through better runtime estimation and the error detection can automatically detect several types of errors.

1 Introduction

Achieving performance in the environment of the Grid is difficult because of the heterogeneity implied by such a Large Scale Distributed System (LSDS) and because the Grid itself is not a stable platform. One of the promises of the Grid is

Dragos Sbirlea · Alina Simion
Computer Science Department, Rice University,
Houston, Texas, USA
e-mail: alina.gabriela.simion@rice.edu,dragos@rice.edu

Florin Pop · Valentin Cristea
University Politehnica of Bucharest, 313 Splaiul Independentei,
060042 Bucharest, Romania
e-mail: {florin.pop@cs.pub.ro,valentin.cristea}@cs.pub.ro

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 171–191.
springerlink.com

reliable operation at a small cost, compared with high-end conventional computer systems, because the reliability could come from software, rather than expensive hardware [1]. Because of its geographical dispersion, Grids are not prone to many kinds of failures that would make other systems unusable, such as power or network failures). The Grid management software can send automatically resubmit jobs to other machines when a failure is detected, but failure detection, which is based on profiling and end-to-end monitoring, turns these features into requirements in order to make the Grid promise or reliability possible.

In a dynamic, heterogeneous environment such as the Grid, monitoring is the first step towards building a reliable system form an unreliable one. An end-to-end monitoring system that takes into account the various possible uses in the Grid environment of the monitored data is a solution for application like satellite images processing. The main focus using monitoring is have a mechanism for performance analysis based on behavior models, in order to handle both current complexity of systems design and collaborative interactions. Validation of the analysis using simulation tools or real testbed infrastructures proves the correctness of proposed solution.

The chapter is structured as follows. Section 2 presents the related work and introduces the gProcess architecture. In section 3 the goals of the proposed monitoring solution are presented. Monitoring system architecture is described in section 4 and the components description is presented in section 5. Section 6 presents the application runtime estimation, process based on monitoring information and very important in workflow execution. Testing and results for the proposed solution is analyzed in section 7. In section 8 we conclude on the presented solution and experimental results.

2 Related Work

The proposed monitoring solution is built on top of a workflow-based satellite image processing engine which is being built in the context of the SEEGRID research project. This engine, named gProcess, decomposes image processing workflows into operators which are then run on clusters [2][3].

The initial architecture of gProcess was design based on client-server application paradigm (see in Figure 1)[12]. A client sends a image processing request specifying a workflow and the appropriate inputs to the gProcess web services (1) through a locally installed client application. The web services separate the workflow in operators; jobs are created and then send by the scheduler to be executed on a gLite worker nodes from a gLite cluster (2). The output follows the reverse path, until it reaches the user [13][14].

Two workflow types are already available for use in gProcess, the Enhanced Vegetaion Index (EVI) [15] and the Difference Drought Index (NDDI) [16].

A workflow instance adds the corresponding input files so that the processing can be started. The workflow works with multi-spectral satellite images. Basic transformations operate on a single spectral band and define operators in a workflow. They

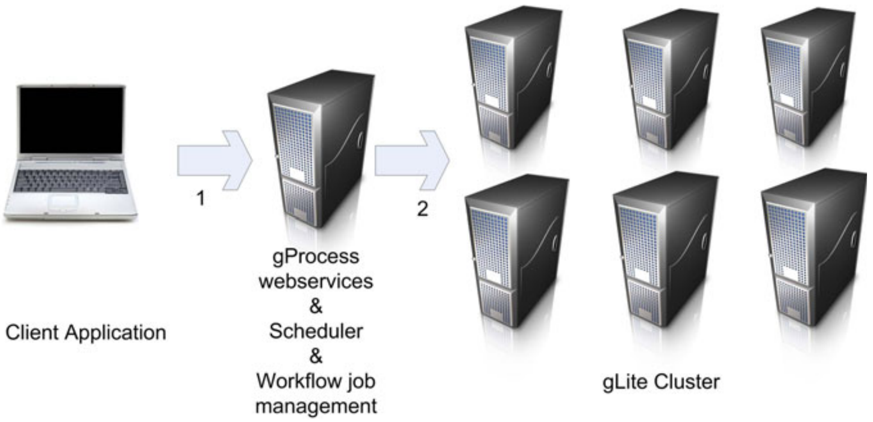


Fig. 1 The initial gProcess architecture; it was changed while implementing the monitoring solution

can compute arithmetic transformations on two images, like the Subtraction (Sub) and Addition (Add) operators, or on one image and a floating point number, like the Multiplication with constant (MultFloat) and Division operators. Some arithmetic transformations require only an image, without a second parameter, like the Complement (Compl) operator. Basic transformations can compute geometric transformations, like rotation, scale and translation, or visualization transformations, such as edge detection or spot detection. These transformations map to operators in gProcess terms, which means they are similar to jobs.

Monitoring Agents using a Large Integrated Services Architecture (MonALISA) [5] is a distributed monitoring system built using a dynamic distributed architecture based on autonomous agents that interact an collaborate in order to perform information gathering and processing. The advantage of such architecture is its scalability. This tool has a good integration with existing monitoring tools and utilities and allows collection of various types of parameters about computational nodes, networks as well as custom application parameters. Based on Java, JINI and web services, MonALISA is used non-stop in more than 350 sites; it's perfectly suited for around the clock operation because it can easily adapt to dynamic changes in the environment. The real-time data visualization feature of MonALISA is an important feature which has proven valuable during the course of this project. A MonALISA repository, which includes a powerful data visualization and analysis engine is used for retrospective data analysis. The repository is built using enterprise java technology running on a Tomcat web server, around a scalable and fast database engine (PosgreSQL). As far as programming language integration, MonALISA allows sensor creation using Java, C, C++, Perl and Python.

3 Goals of the Monitoring Solution

The goals of the proposed monitoring framework are: offer sufficient error management (used for fault recovery), help scheduling improvement (through relevant statistics), do thorough resource utilization accounting and be easily extensible and reusable in another application.

A goal of the monitoring solution is to allow easy visualization of current (real-time) and historical data. Accessibility of these visualization tools is important in the context of the Grid: the more accessible the better. The perfect solution would mean remote access to text based and graphical representations of the data gathered.

It is a primary goal of the monitoring solution to discover, log and announce any condition that might affect the proper functioning of the system. The system should also attempt automatic recovery after an error condition occurs (for example: set a number of retries for a job that was not successful the first time). The monitoring solution should make it easier for the scheduler part of the system to adapt to such conditions.

Another important contribution of the monitoring framework in workflow application concerns scheduling performance. Statistics gathered by the monitoring subsystem can be successfully used to improve the quality of the scheduling [6] [7] [8]. Furthermore, "the quality of the estimated [run] times is essential for the quality of the schedule" [9]. Because it offers real time information on the job runtime in Grid environments, the monitoring system plays an important role in estimating execution time. These estimates need not be very accurate to see an improvement in schedule performance [10].

The monitoring data should be aggregated in such a way as to make it easier for the workflow or application developers to identify bottlenecks and, once identified, to alleviate the problems. In the heterogeneous and rapidly changing context of the Grid, extensibility and reuse are of utmost importance. The solution should be portable to as many operating systems as possible and not be dependent of additional technologies.

4 Monitoring System Architecture

This paragraph describes the improved gProcess architecture, that integrates the monitoring solutions component with the gProcess components.

The proposed, improved gProcess Architecture, consists of Client Application, Web services, Scheduler, Job Management Unit, customized MonALISA Repository, Monitoring Database Server, Automated Error Detection System and Statistics Computing Unit. The client application is the application that requests the execution of a satellite image processing workflow. It can be located anywhere, but must have network access to the GreenView web services, where it sends the request and from where it gets the results back [13]. The gProcess web services are web services built using Java, and deployed in GlassFish Enterprise Application Server and contain scheduling and job management components which decompose the workflow into tasks, schedule the tasks for running on the gLite cluster, checking that all dependencies are met and get the final output.

The MonALISA repository and database server collect the monitoring information generated by the running jobs and store it for later visualization and accounting. Monitoring information generated by the running jobs is collected and stored using a MonALISA repository and database server, for accounting, visualization and post-processing.

The relationship between all these components is shown in the following system information flow diagram (see Figure 2). A client sends a image processing request to the gProcess web services (1). These services separate the workflow in operators; jobs are created and then send by the scheduler to be executed on a gLite worker node from a gLite cluster (2). The worker nodes that execute the jobs (operators) send monitoring data to a MonALISA Repository for visualization over the Internet and also to a database server for extraction of statistical information (3). The automated error detection system and the statistics computing unit query the database and update it to discover the errors and to reflect the updated statistics values (4). The statistics are used by the scheduler to improve the scheduling process (5).

5 Components Description

The system components that required modification and adaptation to the new monitoring solution are: Job Management Unit, Scheduler, Operator and Workflow Engine. New components that were added are: Monitoring Sensors, Monitoring Engine, MonALISA Repository, Monitoring Database Server, Automated Error Detection System, Estimates Computing Unit, Scheduling Costs Wrapper Classes and Notification Service. In this section a description of each of these components, presented (colored) in figure 3, will be offered.

5.1 Monitoring Sensors

Monitoring Sensors are the components that accomplish the actual extraction of system and application parameters. Although MonALISA offers access to numerous and diverse parameters, these are only system parameters, that do not characterize evolution of the running job, but that of the system. Correlating the job and system behavior remains is a difficult task without job parameters.

The proposed implementation contains custom sensors for systems parameters (these are used if MonALISA does not supply a particular system sensor, or if the sensor implementation is not compatible with the operating system used). A more important feature of the monitoring solution is the possibility for the operator creator to implement their own sensors, that may transmit job related information. Correlated with the system sensors information, this allows better results in remote profiling and system tuning using the proposed monitoring solution.

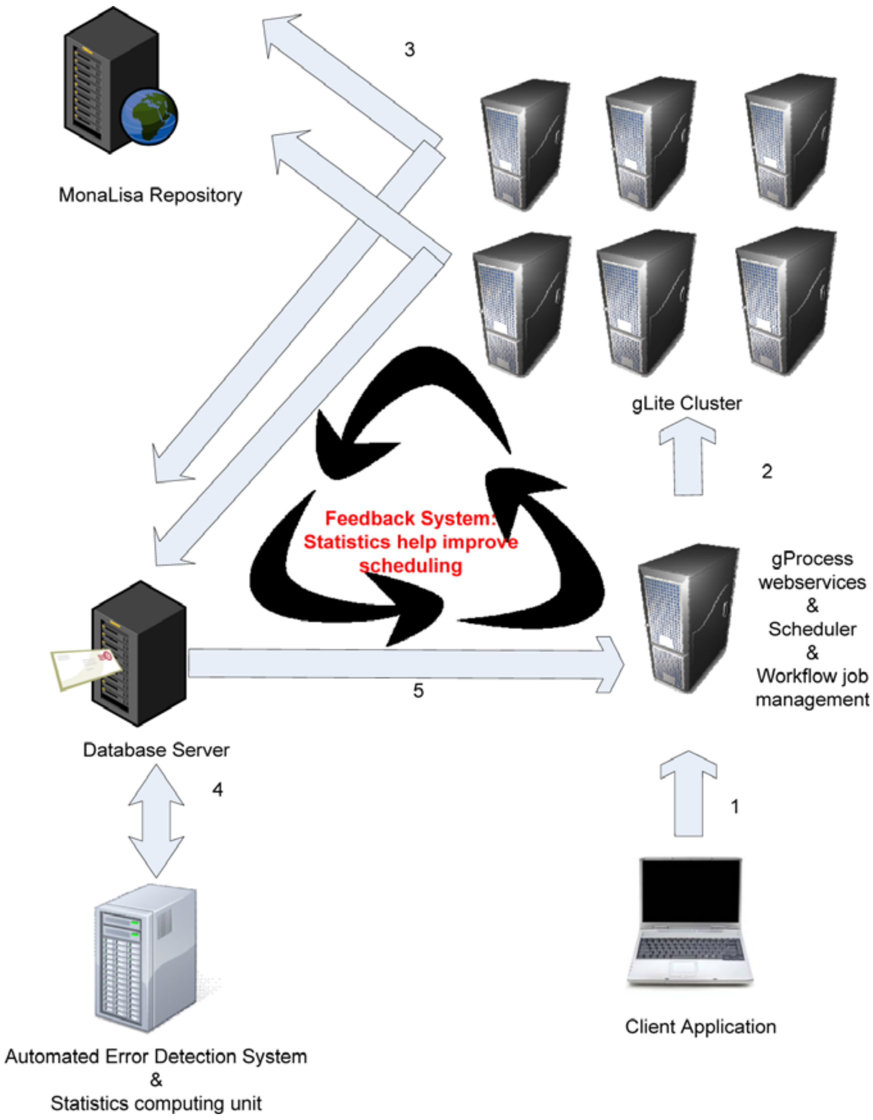


Fig. 2 The improved architecture of gProcess, showing the implemented components of the monitoring system and their integration with the original components. The diagram shows the information flow through the system.

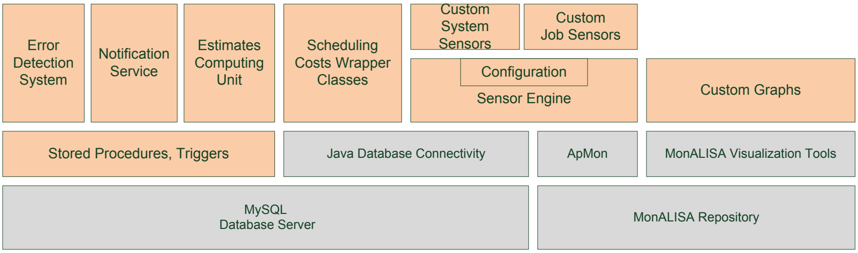


Fig. 3 Monitoring System Components, represented as a stack. The components implemented for the proposed monitoring solution are colored and the external libraries and frameworks used are grayed out.

A small framework that allows for an easy to use, extensible sensors system has been created and is running in the context of the operators that are in fact jobs in a gLite cluster. Because gProcess is developed in Java, the easiest way to interact with it is by writing all code in Java, which was the approach we took. Each operator sent to be executed is wrapped by a special MonitoringOperator object that adds monitoring capabilities and forwards all actual processing done to its wrapped object. Practically, by adding this Monitoring Operator class and, with a small modification in the Operator loading code, monitoring capabilities have been added to a pre-existing application. The goal of this approach was to be minimally invasive in the pre-existing code and keep an elegant design nevertheless.

Because one of the goals of the monitoring solution was extensibility, the actual sensors used are separate Java classes. These are dynamically loaded through Java Reflection API and offer a very powerful extensibility feature to the system. It is to be noted that the current implementation does not even require that these classes implement a certain interface, only that the actual function that does the sensor data retrieval taken no parameters and returns a double (unlike an interface, the function name is not important). Each sensors class, being loaded on demand at runtime can be sent to the worker nodes only when an operator uses it. This creates the possibility of in place upgrades, without need of restart and thus without imposing downtime to the system. In services where the Service Agreement requires a certain uptime, this is especially important.

From the application (job) sensors implemented, the most interesting is the life cycle step sensor, which allows knowing remotely the evolution of the running operator, if it supports this. Such information may be useful in estimating the runtime of that operator, in error checking and debugging (when an operator is stuck in a certain life cycle stage) or in system tuning (the *n*-th stage of operator Add takes up too many resources which shows developers where to optimize that operator).

The life cycle step of Jobs is a custom job (application) sensor type. In the graphic presented in Figure 4, on the left, we see the evolution of an Add operator (life cycle steps 1 to 4) and on the left, a simple operator (a RGB filter) who uses only steps 1 and 3. The values, scale and type of application sensors depend only on the application developer. In the background, the main window of the MonALISA Web

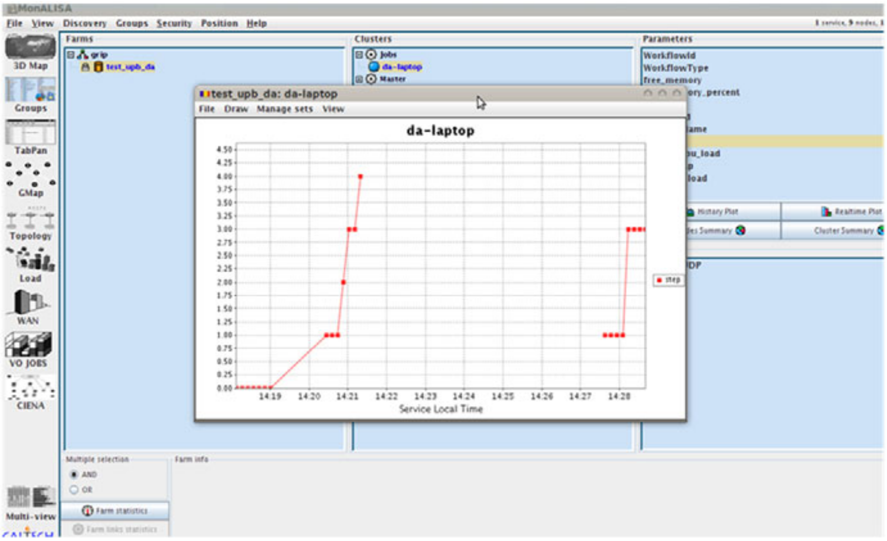


Fig. 4 Monitored life cycle steps

Client is shown; in the right we see the sensor selection list which allows access to the graphs for every sensor offered by the system.

5.2 Monitoring Engine

The Monitoring Engine handles the configuration of monitoring sensors and their arrival at available data repositories. Because each operator might require different sensors, each operator executed comes with a configuration file that indicates the classes and function names of the sensors used. This configuration file can be easily changed, for example to allow the same operator to use different sensors according to which workflow it belongs or to allow the system to enter a debug mode where the monitoring level is increased. These configuration files are in XML format for easy processing by machines and humans and allow setting the interval between consecutive sensor readings.

In the following example, a configuration file for monitoring sensors of operators Sub and Add are displayed; the active sensors are located in a Java class called SystemData (which contains most of the system parameters).The sensor frequency is set to 5 seconds, meaning each 5 seconds, sensors are read and the data obtained is sent to repositories. The sensor name is a string that identifies the values read in the database, it might be different for different operators using the same sensor and can be used as desired by the application administrator. The fragment shows a possible configuration for the Add operator. This configuration is used by gProcess for workflow execution. The sensors used are listed in the operator node. The other operator XML node (with the value of the name attribute "Operator") is used in case the configuration file is sent (possibly by mistake) as configuration for another operator. These values are used as a failsafe (not having them means no monitoring data is sent).

```
<monitoringData frequency="5">
   <operator name="Add" class="SystemData">
      <sensor name="user_cpu_load"
              method="getCpu"/>
      <sensor name="system_cpu_load"
              method="getSysCpu"/>
      <sensor name="used_swap"
              method="getSwapUsed"/>
      <sensor name="free_memory"
              method="getFreeMemory"/>
      <sensor name="free_memory_percent"
              method="getFr[...]"/>
   </operator>
</monitoringData>
```

The Monitoring Engine sends the sensors data to the repositories. To do this, another problem must be solved, and that is assigning correct operator identity. Matching a job with its appropriate sensor data is done based using an operator configuration file which contains unique identifiers for an operator in the workflow (Add-1), the workflow type ("EVI"), the workflow instance to which it belongs (for example "EVI-24"), like in the following:

```
<operator name="Add"
          workflowTypeId="EVI"
          workflowId="EVI-24"
          operatorWorkflowId="Add-1">
```

The workflow for EVI is sent by user to gProcess execution platform. The input data for the workflow operators is accessed by gProcess in Grid using GridFTP protocol.

5.3 ApMon Usage

The Monitoring Engine uses ApMon, a Java library that works in conjunction with a MonALISA farm. ApMon helps to send the parameter values to the MonALISA farm which is installed on the gLite gateway. Because the farm is installed on the cluster gateway, it can access (using the pbsnodes command) system statistics for each node, including CPU and RAM usage. Several default sensors can be used to monitor systems, using ApMon.

5.4 The Choice of Data Repositories

The repositories included with the system are a database running on a MySQL Server and a MonALISA Repository. Although MonALISA is the de-facto standard for large scale monitoring systems, we opted for a hybrid approach because of the difficulty of using MonALISA to compute the statistics we desired. MonALISA is used, in our solution, for visualization of system parameters in general. Because application parameters are in general characteristic to the particular operator that is running, relevant visualization is more complicated to do using the fixed 3-level nesting level offered by MonALISA.

A main argument for not relying only on MonALISA is redundancy. Data is kept in the MonALISA and MySQL databases increasing the data safety. Further arguments for choosing this hybrid approach is that by doing this we obtain independence on the MonALISA Version. To use MonALISA to calculate the required statistics would have meant to directly access the MonALISA database (a clear warning about the dangers of doing this is given by the developers) or to access them through the supplied web services, which could potentially lock the monitoring framework into using a certain MonALISA version. Furthermore, choosing another solution for data visualization is feasible with the current architecture.

Because one of the goals of the system was easy visualization and flexibility, we used a customized MonALISA repository to create our own graphs of the monitoring data gathered. A repository is also useful for controlling the data gathered (relying on data stored in a third-party server, which you cannot control, is not wise). The repository helps to achieve the goal of data visualization. The solution of using MonALISA for this task meant that time is not wasted by writing code that already exists and as an added advantage is the minimal number of bugs introduced (MonALISA code is used 24/7 in many large scale systems and its reliability is proven).

The customization of the default MonALISA repository was done in three different aspects: customizations affecting the interface, security customizations and graphs creation. The most important change was the creation of new graphs. Each graph has a title, a legend showing what signification has each data series and, according to the graph type, some graphs offer a calendar control for the time interval that should be plotted, a selection list o choose the farm that provides the data that is plotted and a possibility to select only certain series to be represented in the actual graph. Few graphs also offer links to related graphs (see Figure 5).

The graphs are structured in three categories, according to their purpose, as shown in the following screenshot of the MonALISA customized menu. Some graphs are used to monitor the sensors form all nodes; these graphs allow visual representation of CPU History, Memory Usage History and combined view of both of the above. The second category of graphs shows the values of the runtime estimates for the available workflows. These show both the real time values (as pie-chart) for these estimates and the history (as x-y charts). The third graphs category is the one that allows visualization of server sensors only; these graphs are useful in server profiling and performance tuning. From the MonALISA perspective, the created graphs are classified as: pie charts (real-time runtime estimates for EVI

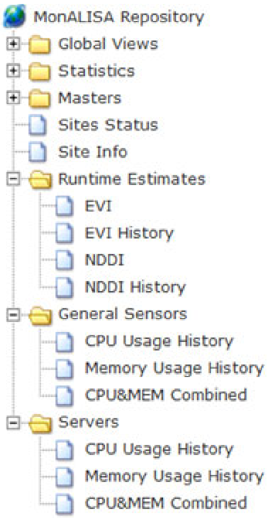


Fig. 5 MonALISA Repository customized menu

workflow), history (CPU, memory usage), real-time (server CPU usage in real-time) and combined (CPU and memory combined).

A important part of the graph creation effort was spent on making the current implementation to be flexible enough to support the further development of the gProcess platform. Here, we find that the servers used (scheduler and web services servers) are discovered dynamically. When adding a new server to the system, there is no need to modify this graph; just start the server monitoring and the data will appear in the servers CPU history graph.

5.5 *Monitoring Database Server*

The second type of data repository used in the solution is a MySQL database server; the server is used to compute statistics later used for scheduling improvement, to identify errors and to more easily search through the monitoring data (without the three level limitation of MonALISA).

Because the statistics use the computation of operator runtime, the database contains a table for logging the start and stop time for all operators, as well as their identification information. When all operators from a workflow instance are finished, the statistics for the workflow instance can be calculated. The single per - workflow statistic is the perceptual runtime estimates for operators in a workflow, described in the next chapter. This estimate is computer using a trigger and stored procedures.

5.6 Automated Error Detection System

Currently, the system detected errors are timeout and job duplicate errors. To mark timeout jobs, the operators whose end time is not set in the corresponding database table after a set amount of time are moved to a special table. The simplest fault tolerance method that can be built upon such error detection mechanism, is the job retry method (the scheduler can attempt to reschedule them). This creates a simple error detection system and allows fault recovery system at the same time. Other types of errors can be added to the same table but specific methods of identification should be devised for each of them. A version of such method is already implemented by gLite but it does not always work, because job errors are not detected by gLite. This is a major advantage for the error detection system, but it further work is needed to identify possible job-related errors and add the detection mechanism to the other error detection database functions.

5.7 Notification Service

The Notification Service is a program, created to be run periodically (maybe as a cron job in Linux environment) which send notifications to the administrator concerning the errors that have been discovered. Now, it supports ending email (plain text and html with images) and is implemented as Java classes.

5.8 Estimates Computing Unit

The Estimates Computing Unit updates the runtime estimates to the latest values after a workflow is finished. This is accomplished through a trigger that checks, on insert, in the table that logs the start and end times of operators, that a workflow has been successfully finished. It uses the previous statistics and the current values obtained from the newly executed workflow to obtain new estimates (statistics) for the next execution of that particular type of workflow. These estimates values are the sent to the MonALISA Repository for storage and visualization.

5.9 Scheduler Costs Wrapper

The Scheduler Costs Wrapper classes allow the scheduler to access the runtime statistics computer by the Statistics Computing Unit and saved in the Monitoring Database Server. These Java classes ease the job of the scheduler by aggregating and simplifying access to the runtime estimates. These are kept in a type safe (generic) hash map and are extracted atomically from the database. The atomicity needs to be maintained in order for the percentages for all operators extracted to have the sum of 100. For the scheduler, this is not an absolute requirement, but worse schedules might still appear in rare cases, which might lead to longer total execution times. This wrapper over the statistics database might also be used as a base class for other scheduler cost computing classes if the scheduling algorithm uses other kind of costs than those provided.

5.10 Goal Related Issues

The generality of the solution was a main concern. The solution was tested inside and outside(in large distributed systems) of the Grid environment, where it performed according to specifications. It is expected that reuse and extensibility are more than enough to assure fast development times for applications that choose the proposed monitoring solution.

6 Runtime Estimation

Previous solutions to assigning a cost to each Grid job involved estimating the actual runtime of a job. However, this is a difficult problem to solve, even if that job has been run already, in the past. These difficulties arise from job parameters being different (an image of a much larger size as input would lead to a larger runtime) or because of the dynamic nature of the Grid environment.

6.1 Using Statistical Data to Estimate Runtime for Jobs

Some batch queuing systems require the user to provide job runtime estimates, instead of trying to statistically compute them. Research has show that these estimates are wildly incorrect, with up to 60% of jobs using less than 20% of the estimated time [11]. Many factors contribute to this discrepancy between estimated time and real job execution time. One such factor might be that many jobs crash immediately as execution starts; the inherent difficulty of estimating runtime and the varying load of the systems used can be another.

A contribution of this chapter is proposing new, relative estimates of the runtime of related jobs, in the context of workflows. The proposed solution offers a operator cost computing component which associates a certain cost to each operator used in a certain position in a workflow. The cost assigned to each operator is an estimate of the runtime of the operator divided by the runtime of the workflow and scaled to 100. This information is computed statistically using data from all instances of that particular operator in that particular type of workflow, that were run in the system.

In general, the cost associated with the execution of an operator in the $N+1$ execution of a certain workflow wf should be estimated using a particularization of the following formula: $Cost_{op}^{\prime wf}(N+1) = f(Cost_{op}^{wf}(1), Cost_{op}^{wf}(2), \ldots, Cost_{op}^{wf}(N))$, where: $Cost_{op}^{wf}(k)$ is the measured cost of the execution of operator named op in the k-th execution of workflow wf in the system. It is not estimated, but computed, because the running time of the operator and its workflow is known. $Cost_{op}^{\prime wf}(N+1)$ is an estimate of the cost of the next operator, used in the scheduling of the op operator in the wf workflow. For example, a cost is associated with the first Subtract operator in the EVI workflow type: $Cost_{Sub-1}^{EVI} = 1.25$ [13].

The initial cost of each operator in a workflow can be set by the programmer, but default values are assigned automatically if this is not done. Default values assign the same cost to each operator in the workflow. Although this might not be a good approximation, the statistical engine will update these parameters to more realistic values as soon as the workflow is run a few times.

Notice that the operator cost is in relation only to the runtime of other operators from the same workflow (dependent on the same inputs) and not to those in other workflow. This is an improvement over other estimation methods, because the relationship between runtime of operators in different workflow types is uncertain. Also, this cost is updated with each new workflow execution, which means the cost is adjusted when changes caused by the dynamic nature of the Grid occur (for example, other type of machines are installed, and they execute some operators faster and other slower; this might happen when the IO bandwidth is decreased, but the FLOPS are increased on the new hardware).

Because the number of executions of a workflow can potentially be very large, using data from all the past executions to calculate an estimate can thus cause performance problems. This is why we preferred to use a simpler formula, one that does not need the cost of every execution of that operator. We have chosen to use only the previous estimate and the last actual execution cost to computer the next estimate of the cost: $Cost_{op}^{\prime wf}(N+1) = f(Cost_{op}^{\prime wf}(N), Cost_{op}^{wf}(N))$.

Because the updated cost is not needed instantly, it is also feasible to use the complete formula and just choose a particular function f. However, the computing cost is not justified, for reasons discussed below.

Table 1 Relative runtime (out of 10) for the first 5 executions of the EVI workflow

Workflow ID	EVI-135	EVI-136	EVI-137	EVI-138	EVI-139
Input Images	Romania	Japan	Italy	France	South America
Images Size	2480*3508	5800*7200	3933*4717	5200*4000	5940*8520
MultFloat-5	1.03	1.06	0.84	1.02	1.06
MultFloat-8	1.03	1.03	0.90	1.02	1.06
Sub-3	1.61	1.41	1.81	1.58	1.58
AddFloat-11	1.26	1.03	0.90	1.02	0.95
Add-9	1.49	1.70	1.94	1.64	1.60
MultFloat-14	1.03	1.03	0.84	1.07	1.08
Add-12	1.49	1.06	0.84	0.96	1.06
Div-15	1.03	1.06	0.84	0.96	1.06

Considering the simplification above, the problem remains to choose the function f so as to balance the following requirements:

- Quick adaptation to workflow pattern change.
- Good resistance to few statistically aberrant results
- Proven to respect the fundamental characteristic that $\sum_{op} Cost(op) = k$ (The sum of the estimated cost of all operators in a single workflow wf, estimated at any moment, should be equal to k, where k is a fixed constant).

The sum of the costs associated to the actual runtime of the operators for a workflow is considered to be k. This is consistent to a cost that represents a percentage of the workflow runtime, if run on a single machine. We propose two potentially good formulas for runtime estimates, considering the previously mentioned restrictions.

The FPE Formula. The first formula considered has a *fixed percentile adjustment rate*:

$$Cost_{op}^{\prime wf}(N+1) = Cost_{op}^{\prime wf}(N) * (1-p) + Cost_{op}^{wf}(N) * p.$$

Proof of Correctness of FPE Formula. For simplification, we will consider $op'_1 \ldots op'_n$ to be the estimated costs for the t execution of a workflow wf. We presume that these operator costs respect the formula $\sum_{op'} Cost(op') = k$. We consider that $op_1 \ldots op_n$ are the actual costs for these operators and we want to prove that $op''_1 \ldots op''_n$ (the estimated costs for the $t+1$ execution of the same workflow), respect the same formula, $\sum_{op''} Cost(op'') = k$. The actual costs, because of the way they are computed, respect the formula $\sum_i Cost(op_i) = k$. Using these terms, the FPE formula can be expressed as follows: $Cost(op''_i) = Cost(op'_i) * (1-p) + Cost(op_i) * p$. By adding the previous relation, for each operator in the considered workflow, we get:

$$\sum_{op''} Cost(op'') = \sum_{op'} Cost(op') * (1-p) + \sum_{op} Cost(op) * p$$

By replacing the sums with their values (presented before), $\sum_{op''} Cost(op'') = k * (1-p) + k * p = k$.

The DPE Formula. The second formula considered has a *decreasing percentile adjustment rate*:

$$Cost_{op}^{\prime wf}(N+1) = Cost_{op}^{\prime wf}(N) * \left(1 - \frac{p}{N}\right) + Cost_{op}^{wf}(N) * \frac{p}{N}.$$

Proof of Correctness of DPE formula. The proof is similar with the proof for FPE, but in this case p is variable and equal with $p' = p/N$. By replacing in the previous proof p with p' we get the desired proof.

6.2 *Implementation of Runtime Estimates*

The estimates are updated using a trigger on the database's table that stores the starting and ending time of operators. When the operator is the last one of its workflow, the percentages of time spent running each operator from that workflow are computed and these values are used to update the runtime estimates. After being updated, the statistics values are sent to the MonALISA Repository for visualization.

7 Testing and Results

The testing of the proposed solution was done on a gLite production cluster from University POLITEHNICA of Bucharest set-up for SEEGRID Project. The cluster is a high performance cluster built around two cores: one with 32 dual-Xeon computers with 2GB RAM and one with 48 P4 HT computers at 3GHz, connected with a Gigabit network. Besides these components there are also storage servers and auxiliary applications, as well as the pre-production cluster (24 P4 computers at 3GHz).

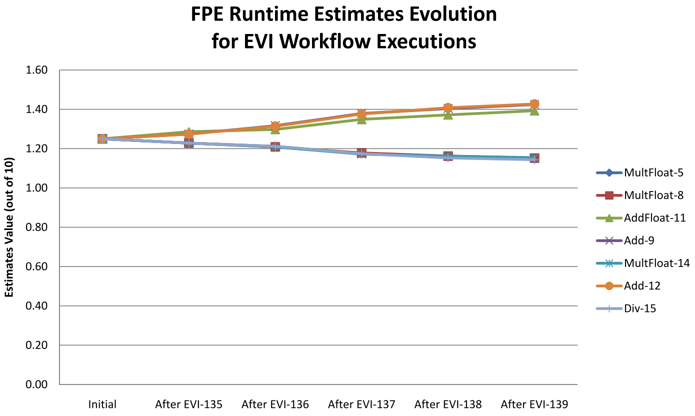


Fig. 6 The FPE estimates get close to their stable values after only a small number of executions of the EVI workflow

7.1 *General Improvements*

After the monitoring solution was installed in the cluster, the first improvement noticed by the development team was that monitoring information about running jobs was available with several (1-10) minutes earlier from the monitoring system than from the gLite middleware. This might lead to an improvement of job performance if, for example, the scheduler policy for resubmitting jobs that fail took into account the error identified by the monitoring system and not those provided by gLite. The monitoring solution proved useful for the developers in identifying a bug that allowed the jobs to remain on the worker nodes indefinitely after completion. Also, it helped to show that the worker nodes were not fully stressed during operator execution, so further optimization of the image processing implementation is useful.

Because the monitoring system is not available at the time, a certain class of monitoring data is not available. For example, monitoring during the decompressing part of the script that contains the job that will be run, before the loading of the Java operator that does the processing is not implemented. Thus, detecting missing Java environment is not possible; a workaround might be found by using other MonALISA bindings, such as those for Python, but this would only mean changing one dependency problem with another. A disadvantage of the runtime estimation system is that it presumes identical machines in the cluster where a certain workflow is run; the system adjusts well when replacing all processing nodes, but does not work properly on heterogeneous clusters.

7.2 Reliability

The implementation used features such as: threads for isolation of the monitoring code, UPD to mitigate timeout errors, MonALISA for scalability and redundancy. The tests shows no sign of decreased reliability and, as expressed above, the monitoring system helped identify irreversible and unknown error types.

7.3 Runtime Estimates Testing

Tests were conducted using instances of the EVI operator showed that the estimates tend to stabilize after a few workflows are executed. This is true for tests are run in one job per worker node situation. Some clusters are configured to accept multiple jobs at the same time, which might lead to bad estimates.

Tests performed on the SEEGRID cluster at night (low load) showed almost constant estimates. Daytime tests (heavy load) lead to variations in run times of over 1000% which mean the data from executions that take place under uncontrolled load should not be used to improve the estimates (we disabled the updating of the estimates when the system load is larger than a set limit). The input images for testing the runtime estimates consisted of satellite images of medium size, mentioned in Table 1 and 2. This tables presents the estimation cost for the first five executions of the EVI workflow.

The FPE estimates, presented in Figure 6 get close to their stable values after only a small number of executions of the EVI workflow. This show that we have a short overhead for estimation step and the costs for workflows could be used successfully in a scheduling system.

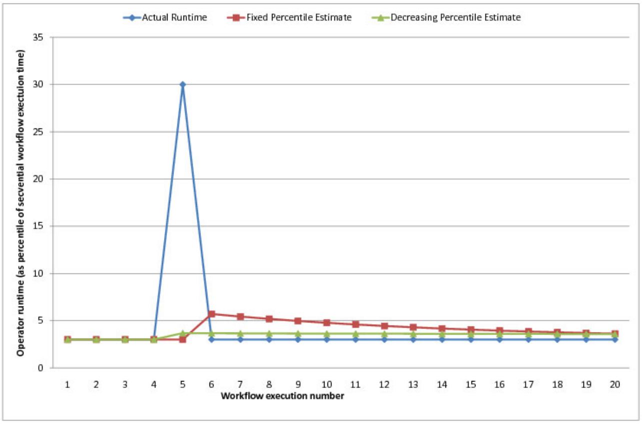


Fig. 7 The evolution of the two proposed estimated of the operator runtime relative to the workflow, in the case of a single execution. DPE has the smallest variation, possibly leading to better scheduling performance than FPE.

Because the test data showed no change in workflow profile, both estimates were quick to adapt to the workflow profile. The DPE ($p = 0.1$) was much slower than FPE, which can be attributed to a low value of p (the influence of the new runtime value is small). A greater value for p would have helped in this case. The DPE tests with $p = 0.3$ confirmed this, but the better adjustment rate proves to be a disadvantage if the first few operators are not representative (typical) of the others, so this makes high values of p usable only if the first executions of newly introduced workflows are run in a controlled environment. Lower p values have the advantage that workflow types can be introduced at any time.

The Fixed Percentile Estimate (FPE) formula has the advantage of a rapid adaptation to changes in the execution profile of a workflow. The disadvantage is that it will not reach a stable level, but will keep alternating, according to the variation of actual execution results (see Figure 7).

The Decreasing Percentile Estimate (DPE) formula leads to better stability considering a stable workflow execution profile (after a number of execution, a new, aberrant execution time will cause less change to the estimate) but this has the disadvantage that if workflow profile do change (an operator execution time changes drastically compared to other operators in the same workflow) adaptation to the new cost will be very slow. Because such changes appear rarely after a workflow is created and it is easy to create another workflow with another name instead of using the same one, the disadvantage becomes less important (see Figure 8).

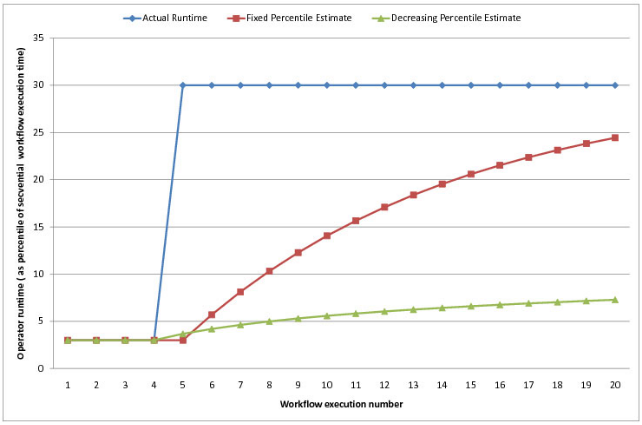


Fig. 8 The evolution of the two proposed estimated of the operator runtime relative to the workflow, in the case of a workflow execution profile change. FPE is the quickest to adapt, possibly leading to better scheduling performance than DPE.

Table 2 The FPE Evolution with $p = 0.1$ for the first five executions of the EVI workflow

	Initial	EVI135	EVI136	EVI137	EVI138	EVI139
MultFloat-5	1.25	1.23	1.21	1.17	1.16	1.15
MultFloat-8	1.25	1.23	1.21	1.18	1.16	1.15
Sub-3	1.25	1.29	1.30	1.35	1.37	1.39
AddFloat-11	1.25	1.25	1.23	1.20	1.18	1.16
Add-9	1.25	1.27	1.32	1.38	1.40	1.42
MultFloat-14	1.25	1.23	1.21	1.17	1.16	1.15
Add-12	1.25	1.27	1.31	1.38	1.41	1.43
Div-15	1.25	1.23	1.21	1.17	1.15	1.14

8 Conclusion

A modern trait in Grid monitoring systems is the increasing number of uses for the collected data. One domain in which this data can have a significant improvement on the performance of Grid application is using the real-time information to compute estimates of runtime which are used to improve scheduling. The proposed estimates proved were successfully used for the workflows tested.

Another such domain is automated error detection systems, which can improve the robustness of Grid by enabling fault recovery mechanisms to be used. Both these aspects can benefit from the particularization of the monitoring system for a workflow-based application: the scheduling performance can be improved through better runtime estimation and the error detection can automatically detect several types of errors. Of a fundamental importance in building such components for a distributed application is using a real-time monitoring framework such as MonALISA, used for scalability.

The importance and value of this project rest not only in drawing the conclusions mentioned above, but also in more concrete facts. The solution is used in the SEE-GRID-SCI project and will be a part of the satellite image processing engine that is being built.

References

1. Zanikolas, S., Sakellariou, R.: A taxonomy of grid monitoring systems. Future Gener. Comput. Syst. 21(1), 163–188 (2005)
2. Bacu, V., Gorgan, D.: Graph Based Evaluation of Satellite Imagery Processing over Grid. In: ISPDC 2008: Proceedings of the 2008 International Symposium on Parallel and Distributed Computing, pp. 147–154. IEEE Computer Society, Washington (2008)
3. Gorgan, D., Stefanut, T., Bacu, V.: Grid Based Training Environment for Earth Observation. In: GPC 2009: Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing, pp. 98–109. Springer, Heidelberg (2009)
4. Simion, B., Leordeanu, C., Pop, F., Cristea, V.: A Hybrid Algorithm for Scheduling Workflow Applications in Grid Environments (ICPDP). In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1331–1348. Springer, Heidelberg (2007)
5. Cirstoiu, C.C., Grigoras, C.C., Betev, L.L., Costan, A.A., Legrand, I.C.: Monitoring, accounting and automated decision support for the alice experiment based on the MonALISA framework. In: GMW 2007: Proceedings of the 2007 workshop on Grid monitoring, pp. 39–44. ACM, New York (2007)
6. Xhafa, F., Carretero, J., Barolli, L., Durresi, A.: Immediate mode scheduling in grid systems. Int. J. Web Grid Serv. 3(2), 219–236 (2007)
7. Streit, A.: A Self-Tuning Job Scheduler Family with Dynamic Policy Switching. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2002. LNCS, vol. 2537, pp. 1–23. Springer, Heidelberg (2002)
8. Xhafa, F., Abraham, A.: Computational models and heuristic methods for Grid scheduling problems. Future Gener. Comput. Syst. 26(4), 608–621 (2010)
9. Iordache, G., Boboila, M., Pop, F., Stratan, C., Cristea, V.: A decentralized strategy for genetic scheduling in heterogeneous environments. Multiagent Grid Syst. 3(4), 355–367 (2007)
10. Armstrong, R., Hensgen, D., Kidd, T.: The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions. In: HCW 1998: Proceedings of the Seventh Heterogeneous Computing Workshop, p. 79. IEEE Computer Society, Washington (1998)
11. Cirne, W., Berman, F.: A comprehensive model of the supercomputer workload. In: WWC 2001: Proceedings of the Workload Characterization, IEEE International Workshop on WWC-4, pp. 140–148. IEEE Computer Society, Washington (2001)
12. Gorgan, D., Stefanut, T., Bacu, V.: Grid Based Training Environment for Earth Observation. In: Abdennadher, N., Petcu, D. (eds.) Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing, Geneva, Switzerland, May 04-08. LNCS, vol. 5529, pp. 98–109. Springer, Heidelberg (2009)

13. Mihon, D., Bacu, V., Meszaros, R., Gelybo, G., Gorgan, D.: Satellite Image Interpolation and Analysis through GreenView Application. In: Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, February 15-18. CISIS, pp. 514–519. IEEE Computer Society, Washington (2010)
14. Bacu, V., Gorgan, D.: Graph Based Evaluation of Satellite Imagery Processing over Grid. In: Proceedings of the 2008 International Symposium on Parallel and Distributed Computing, July 01-05. ISPDC, pp. 147–154. IEEE Computer Society, Washington (2008)
15. Zhang, L., Furumi, S., Muramatsu, K., Fujiwara, N., Daigo, M., Zhang, L.: A new vegetation index based on the universal pattern decomposition method. Int. J. Remote Sens. 28(1), 107–124 (2007)
16. Kendall, W., Glatter, M., Huang, J., Peterka, T., Latham, R., Ross, R.: Terascale data organization for discovering multivariate climatic trends. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC 2009, Portland, Oregon, November 14-20, pp. 1–12. ACM, New York (2009)

Scaling Relations of Data Gathering Times in an Epidemically Data Sharing System with Opportunistically Communicating Mobile Sensors

Akihiro Fujihara and Hiroyoshi Miwa

Abstract. We investigated data gathering time in an epidemically data sharing system with opportunistically communicating mobile sensors. We proposed a stochastic process of the system where N sensors moved randomly and independently on the $d-$dimensional square grid with size L and when meeting opportunistically at the same position on the grid, the sensors shared and stored all possessing data epidemically. We focused on three data gathering times, that is, latency times that (1) at least one sensor collects all (2) every sensor collects at least one common data (3) every sensor collects all. As a result, we found that in general the complementary cumulative distribution functions of these times decay exponentially in their asymptotic regions. We also examined a decay speed, which is also called relaxation time, of the exponential decay numerically with varying d, L, and N. Finally we showed scaling relations of the relaxation times. We think that these relations are useful for estimating the minimum required number of sensors to collect data within a certain short period of time when the sensors are densely covered on the system.

1 Introduction

In recent years, wireless communication technologies have been developed remarkably, during which styles of information communications have been gradually shift from fixed access points to mobile devices. Many researchers studying on Mobile Ad-hoc Network (MANET), Delay- and Disruption-Tolerant Network (DTN), and

Akihiro Fujihara · Hiroyoshi Miwa
Graduate School of Science and Technology, Kwansei Gakuin University,
2-1 Gakuen Sanda Hyogo 669-1337, Japan
e-mail: {afujihara,miwa}@kwansei.ac.jp

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 193–206.
springerlink.com

Mobile Opportunistic Network have been trying to find next-generation wireless communication networks between mobile peers, such as networks connecting human carried mobile devices (smart phones, portable game devices) and vehicles (cars, buses, trains). In such networks, for the uncertainty of sensor's mobility or for the opportunistic communications, each sensor is in need of a store-and-forward routing technique for solving the delay and disruption of wireless connections, and also has autonomously controlled routing protocols for scalability.

Many routing protocols in DTN have been proposed [1]. Some ones are based on flooding to make the best use of few connection opportunities, called epidemic routing [2], which each peer disseminates all the possessing data to neighboring peers. Although the epidemic routing can transmit packets to destinations with the highest possibility, packets are overly replicated and there finally will be full of unnecessary packets in the network. To solve this problem, Spray and Wait routing [3] in which a wait phase to avoid replicating needless packets is introduced periodically and MaxProp [4] which makes a priority ranking of data and based on the ranking, data to drop are selected are also proposed as the representative examples of efficient epidemic routing.

For an application, we can use the above challenged networks as a data gathering system for tracking positions with sensors. There are some specific examples. ZebraNet [5, 6] where position data of zebras in a savanna are collected by GPS sensors and the collected data are transmitted by the combination of long-range single-hop routing and history-based short-range multi-hop routing and Shared Wireless Infostation Model (SWIM) [7] where position data of whales under the sea are collected by speed and direction sensors and the data are transmitted wirelessly based on epidemic routing were known and used for biological researches. By introducing wireless networks between sensors, both the networks are expected to increase the collection rate of distributed data tracked and stored in each sensor.

We were inspired by these data gathering systems and had an interest in a general property of data transmission between mobile sensors under the epidemic routing protocol. After waiting for a while, for example, as long as sensors can meet each other, it is certain that data scattered in the system can be gathered into nearby one sensor. However, it is not clear how much time it will take until gathering all the data is completed. As far as we knew, there was no study on data gathering time. As our first step, we focus on the data gathering time and investigate how their distribution functions are affected by spatial factors (dimension, system size, boundary effect) and the number of sensors by analysing a simple stochastic process where multiple random walkers exchange data with each other.

2 Related Works

There exist some studies on inter-contact time [8, 9, 10] which was defined by an interval time between two consecutive contacts of a pair of sensors. In these studies, experiments with wireless communications by Bluetooth, Wifi, and Zigbee were carried out to collect data of inter-contact time. As a result, they showed that the complementary cumulative distribution of inter-contact time universally obeys the power-law distribution in the middle time range and the exponential distribution at the tail. When focusing attention on the relative distance between two sensors, the inter-contact time is essentially identical to the recurrence time or the first-passage time which is defined by the time that a diffusing sensor comes back or first reaches a specific site. This result was simply explained by first-passage processes [11] and the exponential decay at the tail resulted from the effect of system boundaries.

There were also studies on data transfer time which was defined by a duration time to transfer data to a certain fixed access point from a mobile peer. Data Mules [12] and Epidemic Routing for Sparsely Populated Mobile Ad Hoc Networks [13] are the examples. They showed that the latency distribution of data transmission obeyed the exponential distribution. This result was also related with the first-passage time.

In our previous studies, we investigated the cumulative distribution function of data gathering time by proposing a simple stochastic model with randomly moving sensors on infinite square grid space [14]. We found that the distribution generally obeys a power-law type distribution whose asymptotic exponent depended only on the number of sensors in the system. We also studied the impact of the boundedness of the square grid space or system boundaries on the data gathering time in a limited case of the periodic boundary condition. We found that the distribution function of data gathering time became the exponential distribution at the tail [15]. In this chapter, we explain more details on impacts on the boundedness of the system especially reporting with the case of the reflection boundary condition and summarize our results.

3 Modelling of the Stochastic Process

3.1 Model Description

We define a system with N mobile sensors on the $d-$dimensional square grid ($d = 1,2,3$), i.e., $\mathbf{x} = (x_1,\cdots,x_d)$ with $0 \le x_i < L$ for $i = 1,\cdots,d$, where L is the size of the system. As the boundary condition of the system, we consider two cases: *periodic* and *reflection* boundary conditions. Each sensor is initially positioned at $\mathbf{z}_j(t = 0) = (x_{1j},\cdots,x_{dj})$ $(0 \le x_{ij} < L,\ 1 \le j \le N)$, where t is time. We assume that the initial positions of sensors are different, i.e., $\mathbf{z}_i(0) \neq \mathbf{z}_j(0)$ $(1 \le i < j \le N)$. Also, a sensor indexed by j initially holds its own original data $\{I_j\}$ each other $(1 \le j \le N)$. All the sensors, at each time step, take simple symmetric random walks meaning that the sensors randomly and synchronously move to one of the nearest neighbor positions on the grid with equal probability. When more than one sensors happen to come to the same position, they share and store all the data they hold epidemically. If, for example, one sensor holding a set of data $\{I_1,I_2\}$ and another sensor holding $\{I_2,I_3\}$ meet each other, the resulting set of data for both of them becomes $\{I_1,I_2,I_3\}$ at the next time step. For simplicity, we also assume that the number of sensors and data are in total N and the number does not change in the whole processes.

This modelling assumes that the transmission range of data is the same order as the migration distance of the sensors within a unit time step and is sufficiently smaller than the system size L.

3.2 Data Gathering Times and Their Distribution Functions

Second, we consider the following three data gathering times, i.e., the latency times that

1. *at least one sensor* completes the collection of *all the data* in the system : $t = T_1$,
2. *every sensor* collects *at least one common data* in the system : $t = T_2$
3. *every sensor* completes the collection of *all the data* in the system : $t = T_3$.

By definition, $T_3 \geq T_1, T_2 (> 0)$ holds for $N \geq 2$, where the equality holds in $N = 2$. Note that the magnitude relation between T_1 and T_2 is not trivial. Although $T_1 = T_2$ always holds in $N = 2, 3$, we can easily give examples of the processes that either $T_1 \leq T_2$ or $T_1 > T_2$ holds when $N \geq 4$.

Around the times T_1 or T_2, the whole data in the system are beginning to gather in the sensors, that is, the lower bound of data gathering time is characterised by T_1 or T_2. The data sharing process is terminated at $t = T_3$ because no more data transfer between sensors occurs after this, which means that T_3 gives the upper bound of data gathering time.

Since all the sensors move randomly, the data gathering times are described by stochastic variables. Therefore, we define (complementary cumulative) distribution functions of the data gathering times as follows.

$$\bar{F}_i(t) = P(T_i > t), \tag{1}$$

for $i = 1, 2, 3$.

4 Asymptotic Distribution Functions of Data Gathering Times

In this section, we show asymptotic behaviors of the distribution functions of data gathering times defined at Sec 3. In the number of sensors $N = 2$, by definition, all the three times are equivalent, i.e., $T_1 = T_2 = T_3$. Also, the data gathering time exactly coincides with the first-passage time [11]. Theoretical results from the first-passage time are useful to understand the property of data gathering time. For this reason, this section is divided by two parts: We first introduce the theoretical results of the first-passage time and second show and explain numerical results in $N \geq 3$.

4.1 Theoretical Results in $N = 2$

In the periodic boundary condition, the topology of the system becomes the $d-$dimensional torus $\mathbf{Z}_L^d$. Here, we introduce the following theorem known for the distribution function of first-passage time in $\mathbf{Z}_L^d$ [16].

Theorem 1. *The distribution function of first-passage time in $d-$dimensional torus $\mathbf{Z}_L^d$ asymptotically follows the exponential decay.*

$$\bar{F}_{fp}(t;d,L) \equiv P(T_{fp} > t) \sim \exp(-t/\tau_{fp}(d,L)), \tag{2}$$

where τ_{fp} is called a relaxation time representing the decay speed of the exponential decay. Furthermore, the relaxation time is described by the following function of d and L.

$$\tau_{fp}(d=1,L) \sim L^2, \tag{3}$$

$$\tau_{fp}(d=2,L) \sim L^2 \log L, \tag{4}$$

$$\tau_{fp}(d \geq 3,L) \sim L^d. \tag{5}$$

It is interesting to note that there is no influence of the initial positions of sensors on the *asymptotic* distribution, which comes from the strong Markov property of discrete random walk [17]. This theorem immediately follows that the distribution function of data gathering time in $N = 2$ obeys the exponential function asymptotically.

$$\bar{F}_i(t;d,N=2,L) = \bar{F}_{fp}(t;d,L), \tag{6}$$

for $i = 1,2,3$. According to this result, we can define relaxation times of T_i as $\tau_i(d,N,L)$ for $i = 1,2,3$, which corresponds to the relaxation time of first-passage time, i.e.,

$$\tau_i(d,N=2,L) = \tau_{fp}(d,L), \tag{7}$$

for $i = 1,2,3$. We checked by Monte Carlo simulations that these results was correct in general. Although there is no exact theoretical result known for the reflection boundary condition, we also confirmed numerically that it satisfies the same results in the distribution function of data gathering time and its relaxation time.

4.2 Numerical Results in $N \geq 3$

Since no theoretical result was found in $N \geq 3$, we checked numerically the asymptotic behaviors. We found that all the distribution functions of data gathering times T_i for $i = 1,2,3$ followed the exponential decay both the periodic and reflection boundary conditions as well as the case in $N = 2$.

$$\bar{F}_i(t;d,N,L) \equiv P(T_i > t) \sim \exp(-t/\tau_i(d,N,L)), \tag{8}$$

for $i = 1,2,3$, where $\tau_i(d,N,L)$ is the relaxation time of the exponential distribution. We observed that the distribution functions $\bar{F}_1$ and $\bar{F}_2$ were quite similar in the whole time range. We also observed that the range of $\bar{F}_3$ was almost twice larger than that of $\bar{F}_1$ and $\bar{F}_2$. This reason can be roughly explained as follows under the condition that T_1 is similar to T_2 in distribution. First of all, at least one sensor has data of all sensors at T_1. Here, suppose that all the data is one data block. After this, the sensor having the data block disseminates it to the other sensors by spending more time T_2 because T_2 is the time that (at least) one data block is shared between all the sensors. Therefore, $T_3 \simeq T_1 + T_2$ is roughly satisfied in distribution.

Selected simulation results in $d = 1,2,3$ with the reflection boundary condition are illustrated in Figs. 1. As can be seen, the relaxation times varies with the number of sensors N. As well as the case in $N = 2$, these times also depend on d and L. In the next subsection, we will consider the details of these dependencies.

5 Numerical Results in the Relaxation Times

5.1 *Dependence on* L

We plotted values of relaxation times which were given by the least-square fitting (LSF) of the tails of the distribution functions numerically obtained in 4.2 with varying L in $d = 1,2,3$ with the periodic and reflection boundary conditions. In general, the relaxation times increase with the increase of L because the larger the system size is, the more space the mobile sensors can possibly escape from the others. We considered the relationship between the system size L and the relaxation time obtained by LSF. Selected simulation results are illustrated in Figs. 5.1. We found that as you can see in the figures the linear relationship was satisfied between T_i and L^2 in $d = 1$, $L^2 \log L$ in $d = 2$, L^3 in $d = 1$ for $i = 1,2,3$ and it holds independent of the details of N and boundary conditions. Therefore, the dependence on L in $N \geq 3$ is as well as that in $N = 2$, i.e.,

$$\tau_i(d = 1,N,L) \sim L^2, \tag{9}$$

$$\tau_i(d = 2,N,L) \sim L^2 \log L, \tag{10}$$

$$\tau_i(d = 3,N,L) \sim L^3, \tag{11}$$

for $i = 1,2,3$. We also performed simulations with changing the initial positions of sensors. It seemed that the asymptotic values of relaxation times quite weakly varied with the change of the positions, but we could not distinguish this result from numerical errors by Monte Carlo simulations because the correlation coefficient of LSF was close to one. Therefore, we may conclude that τ_i is independent of the initial positions of sensors. We also found that all the proportionality coefficients of the LSF lines increase with the increase of N. Next, we pay attention to the details of the dependence of the relaxation times on N.

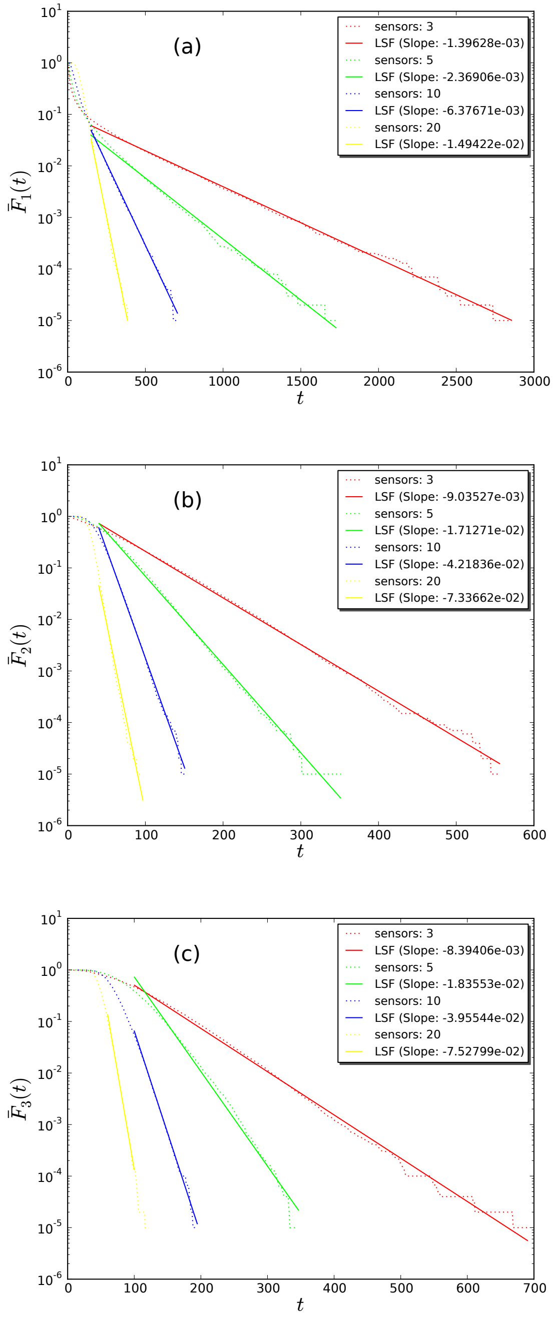


Fig. 1 Distribution functions of data gathering time (a) T_1 in $d = 1$, (b) T_2 in $d = 2$, and (c) T_3 in $d = 3$, where $N = 3, 5, 10, 20$, $L = 50$ with the reflection boundary condition. Numerical results are plotted with dotted lines and their least-square fittings (LSF) are with solid lines.

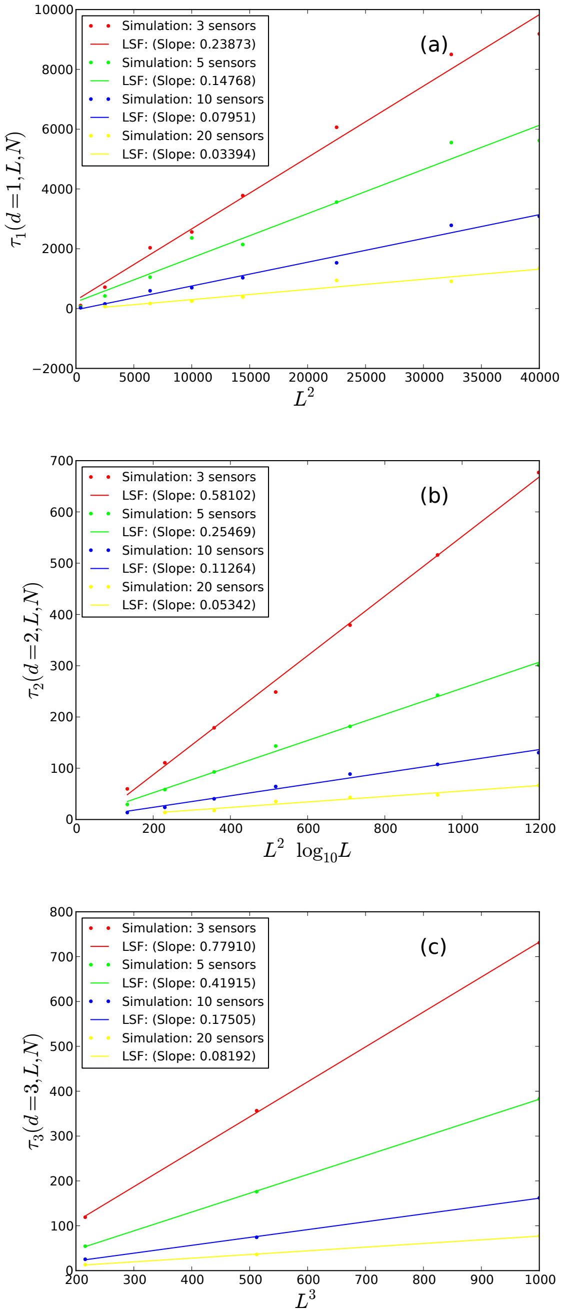


Fig. 2 Relations between the system size L and the relaxation time (a) τ_1 in $d=1$, (b) τ_2 in $d=2$, and (c) τ_3 in $d=3$ with $N=3,5,10,20$ and the reflection boundary condition. Numerical results are plotted with dots and the least-square fittings (LSF) are with solid lines.

5.2 Dependence on N

Here, we refine relaxation times $\tilde{\tau}_i$ from τ_i by removing the dependence on L as

$$\tilde{\tau}_i(d=1,N) \equiv \tau_i(d=1,N,L)/L^2, \tag{12}$$

$$\tilde{\tau}_i(d=2,N) \equiv \tau_i(d=2,N,L)/L^2 \log L, \tag{13}$$

$$\tilde{\tau}_i(d=3,N) \equiv \tau_i(d=3,N,L)/L^3. \tag{14}$$

for $i=1,2,3$. The dependence of $\tilde{\tau}_i$ on N is illustrated in Figs. 3. From the numerical results, we found that $\tilde{\tau}_i$ has a power-law behavior on N in both periodic and reflection boundary conditions.

$$\tilde{\tau}_{ij}(d,N) = 10^{\beta_{ij}(d)} N^{-\alpha_{ij}(d)}. \tag{15}$$

for $i=1,2,3$ and $j=p,r$ (p and r mean the *periodic* and *reflective* boundary conditions, respectively), where α_{ij} and β_{ij} are parameters, which are calculated numerically and summarised in Table 1 and 2.

Table 1 Values of $\alpha_i(d)$ and $\beta_i(d)$ for $i=1,2,3$ in $d=1,2,3$ with the periodic boundary condition

	$d=1$	$d=2$	$d=3$
$\alpha_{1p}(d)$	0.80671	1.25417	1.29942
$\beta_{1p}(d)$	−0.60581	0.19197	0.50488
$\alpha_{2p}(d)$	0.92328	1.23965	1.29327
$\beta_{2p}(d)$	−0.52105	0.18481	0.50262
$\alpha_{3p}(d)$	0.87561	1.10336	1.19514
$\beta_{3p}(d)$	−0.42740	0.15467	0.51286

Table 2 Values of $\alpha_i(d)$ and $\beta_i(d)$ for $i=1,2,3$ in $d=1,2,3$ with the reflection boundary condition

	$d=1$	$d=2$	$d=3$
$\alpha_{1r}(d)$	1.05968	1.12980	1.33444
$\beta_{1r}(d)$	0.09338	0.22472	0.50870
$\alpha_{2r}(d)$	1.17541	1.12554	1.32724
$\beta_{2r}(d)$	−0.01007	0.22776	0.50394
$\alpha_{3r}(d)$	0.99833	0.93315	1.19403
$\beta_{3r}(d)$	−0.00491	0.14992	0.45558

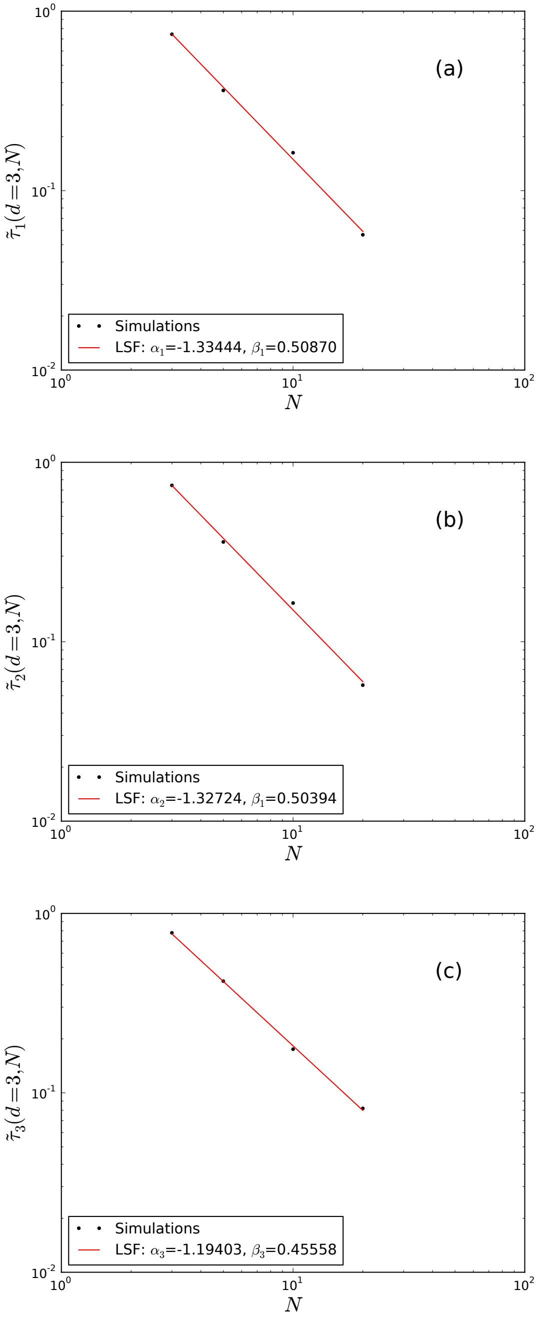


Fig. 3 Relations between the number of sensors N and the relaxation time (a) $\tilde{\tau}_1$, (b) $\tilde{\tau}_2$, and (c) $\tilde{\tau}_3$ in $d = 3$ with the reflection boundary condition. Numerical results are plotted with dots and the least-square fittings (LSF) are with solid lines.

5.3 Scaling Relations of Relaxation Times

To sum up the numerical results shown in the above subsections, the relaxation times τ_i for $i = 1,2,3$ have the following scaling relations with L and N in $d = 1,2,3$.

$$\tau_{ij}(1,N,L) = 10^{\beta_{ij}(1)} N^{-\alpha_{ij}(1)} L^2, \tag{16}$$

$$\tau_{ij}(2,N,L) = 10^{\beta_{ij}(2)} N^{-\alpha_{ij}(2)} L^2 \log L, \tag{17}$$

$$\tau_{ij}(3,N,L) = 10^{\beta_{ij}(3)} N^{-\alpha_{ij}(3)} L^3, \tag{18}$$

for $i = 1,2,3$ and $j = p,r$. Finally, we will consider the application of these scaling relations. Since the exponential decay of the distribution is just an asymptotic property, the average of data gathering time is not necessarily equal to the decay speed (equivalently the relaxation time). In general, the distribution of data gathering time is divided into two ranges of time: One is a power-law range where system boundaries have no influence on the sensor's movement [14], the other is a exponential range where the boundaries become a large influence on the movements, which we consider in [15] and this chapter. We observed that if the sensors were sparsely populated on the system and far from boundaries, then the power-law range broadened and resultingly the exponential range shrank and, on the other hand, if the sensors were densely populated and near to the boundaries, then inversely the exponential range broadened and the power-law range shrank. From this observation, we found that the whole range of the distribution of data gathering time was well fitted by the truncated power-law distribution [18] described by

$$\bar{F}_i(t) = \begin{cases} (t_0/t)^{s_i} & \text{if } (t_0 \le) t \le t_c \\ (t_0/t_c)^{s_i} \exp\left(-\frac{t-t_c}{\tau_i}\right) & \text{if } \quad t > t_c \end{cases} \tag{19}$$

for $i = 1,2,3$, where s_i is the power-law exponents given by the asymptotic behavior in the unbounded space, τ_i is the relaxation time, $t_c = s_i \tau_i$ is around the critical point where the power-law curve turns into the exponential one, and t_0 is the lower bound of the distribution.

We also found that when the number of sensors was sufficiently large such that the sensors was densely populated in the system, the distribution of data gathering time approaches the exponential distribution in the whole range of time, i.e., $t_c \to 0$ and accordingly the relaxation time τ_i for $i = 1,2,3$ was close to the average of data gathering time. In such the case, data gathering is accomplished until $t \simeq \tau_i = const$ with probability $1 - e^{-1} \simeq 0.63$. By using Eqs. (16)-(18), we can inversely calculate the minimum required number of sensors N^* as

$$N^*(d = 1, L, T_i, j) = 10^{\beta_{ij}(1)/\alpha_{ij}(1)} \tau_i^{1/\alpha_{ij}(1)} L^{2/\alpha_{ij}(1)}, \tag{20}$$

$$N^*(d = 2, L, T_i, j) = 10^{\beta_{ij}(2)/\alpha_{ij}(2)} \tau_i^{1/\alpha_{ij}(2)} (L^2 \log L)^{1/\alpha_{ij}(2)}, \tag{21}$$

$$N^*(d = 3, L, T_i, j) = 10^{\beta_{ij}(3)/\alpha_{ij}(3)} \tau_i^{1/\alpha_{ij}(3)} L^{3/\alpha_{ij}(3)}, \tag{22}$$

for $i = 1,2,3$ and $j = p,r$. Since all the value of $\alpha_{ij}(d)$ is around one, furthermore we can estimate the order of N^* as

$$N^*(d=1) \sim L^2, \tag{23}$$

$$N^*(d=2) \sim L^2 \log L, \tag{24}$$

$$N^*(d=3) \sim L^3, \tag{25}$$

These relations indicate that in order to gather data within a certain short period of time $t \simeq \tau_i = const$, we need to cover the system with the sensors to set up the environment that wireless communications occur frequently. This is a reasonable consequence in the scheme of the system of sensor and ad-hoc networks.

6 Conclusion

We investigated the stochastic process of epidemically data sharing systems with opportunistically communicating mobile sensors inspired by biological data gathering systems, such as ZebraNet and SWIM. We considered the model on $d-$dimensional square grid with size L with N randomly moving sensors having its own data. Two boundary condition of the system was considered: periodic and reflection. We analysed the basic properties of the complementary cumulative distribution functions of three data gathering times: the duration of time before at least one sensor completed to gather all the data T_1, the duration of time before every sensor collected at least one common data T_2, and the duration of time before every sensor completed to gather all the data T_3.

In $N = 2$, data gathering time was equivalent to the first-passage time, and furthermore with the periodic boundary condition, i.e., when the space was the torus $\mathbf{Z}_\mathbf{L}^\mathbf{d}$, there was the theorem to give the exact shape of the distribution functions. Based on this theoretical results, we numerically examined the distribution functions of data gathering times with varying d, L, N, and initial positions of sensors. We found that the distribution function $\bar{F}_i(t)$ for $i = 1,2,3$ generally obeys the exponential decay in the asymptotic region of time. We observed numerically that $\bar{F}_1(t)$ and $\bar{F}_2(t)$ had the very similar behavior and the range of time in $\bar{F}_3(t)$ was approximately twice larger than that in the others. We also found that the relaxation time of the distribution is independent of the initial positions of sensors because of asymptotic nature, but does depend on d, L, and N. We derived the scaling relations of the relaxation times in $d = 1,2,3$. This result was applicable to estimating the minimum required number of sensors to gather data within a certain short period of time especially when sensors were densely populated in the system.

In our model, for simplicity, we assume that data which sensors have do not change in time, but our analysis is valid even when the data can change. In this case, data gathering times can give a waiting time until data logged in the past is going to be shared, which is useful in data sharing systems by epidemic routing. Even though the main application will be considered in 2D systems, we also give the results in 1D (for the application in pseudo one-dimensional passages in buildings and streets) and 3D (for the application in three-dimensional sky and space). We think that our results are also applicable to a data sharing system of local contents (micro-blogging, pictures, videos) through mobile opportunistic communications between portable devices.

For our future works, we are preparing numerical simulations that sensors have different mobilities, such as random waypoint and Levy flight models to understand the influence of mobilities on data gathering times. We are also considering the stochastic process run on some graphs, such as complete graphs, tree graphs, and partially defected grids.

Acknowledgements. This work was partially supported by the Japan Society for the Promotion of Science through Grants-in-Aid for Scientific Research (S) (18100001) and (C) (20500080).

References

1. Zhang, Z.: Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. IEEE Communications Surveys and Tutorials 8(1), 24–37 (2006)
2. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Technocal Report CS-2000-06, Department of Computer Science, Duke University (2000)
3. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In: Proc. of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (2005)
4. Brugess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: Routing for vehicle-based disruption-tolerant networks. In: Proc. IEEE INFOCOM (2006)
5. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.-S., Rubenstein, D.: Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: ASPLOS-X Conference (2002)
6. Zhang, P., Sadler, C.M., Lyon, S.A., Martonosi, M.: Hardware Design Experiences in ZebraNet. In: Proc. of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, USA (2004)
7. Small, T., Haas, Z.J.: The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way). In: MobiHoc 2003, Annaplis, Maryland, USA (2003)
8. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. University of Cambridge, Computer Lab Technical Report, UCAM-CL-TR-617 (2005)
9. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of Human Mobility on Opportunistic Forwarding Algorithms. IEEE Transactions on Mobile Computing 6(6) (2007)
10. Karagiannis, T., Boudec, J.-Y.L., Vojnovic, M.: Power law and exponential decay of inter contact times between mobile devices. Technical Report MSR-TR-2007-24, Microsoft Research (2007)
11. Redner, S.: A Guide to First-passage Processes. Cambridge (2007)
12. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In: Proc. of the First International Workshop on Sensor Network Protocols and Applications, pp. 30–41 (2003)

13. Matsuda, T., Takine, T.: (p,q)-Epidemic Routing for Sparsely Populated Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications 26(5) (2008)
14. Fujihara, A., Miwa, H.: Efficiency Analysis on an Information Sharing Process with Randomly Moving Mobile Sensors. In: 2008 International Symposium on Applications and The Internet (SAINT 2008), pp. 241–244 (2008)
15. Fujihara, A., Miwa, H.: Relaxation times of information gathering in an epidemically information sharing process with randomly moving sensors in a bounded space. In: 2009 the 2nd International Workshop on Information Network Design, pp. 360–365 (2009)
16. Aldous, D., Fill, J.A.: Reversible Markov Chains and Random Walks on Graphs. Continuous State, Infinite State and Random Environment, ch. 13, p. 21 (2001), http://www.stat.berkeley.edu/~aldous/RWG/book.html
17. Grimmett, G., Stirzaker, D.: Probability and Random Processes, 3rd edn., Oxford (2001)
18. Fujihara, A.: ZebraNet and its Theoretical Analysis on Distribution Functions of Data Gathering Times. In: The 2nd International Workshop on Moble Opportunistic Networking (MobiOpp 2010), Pisa, Italy (2010)

Multilaterally Secure Ubiquitous Auditing

Stefan G. Weber and Max Mühlhäuser

Abstract. Tracking information of individuals is a useful input to many Ubiquitous Computing (UbiComp) applications. Consider the example of a smart emergency management application: once mobile first responders are continuously tracked, a precise and safe coordination of rescue missions is possible, and also mission logs can be created for audit purposes. However, continuously tracking users and storing the data for later use is often in conflict with individual privacy preferences. This may ultimately lead to the non-acceptance and rejection of these new technologies by their users. In order to reconcile privacy and accountability requirements in location tracking systems, we introduce and evaluate the approach of using auditing mechanisms on two levels. We illustrate that, by employing carefully designed cryptographic mechanisms for selective pseudonym linkability based on efficient techniques of secure multiparty computation, it is possible to balance the conflicting interests to a certain extent. Our work, motivated by and applied to smart emergency management systems, is a step towards the realization of multilaterally secure and thus multilaterally acceptable UbiComp systems supporting collaborative work.

1 Introduction

Creating ICT systems to foster and support collaborative work is a complex issue. Inspired by Mark Weiser's vision of Ubiquitous Computing (UbiComp) which proclaims ”a powerful shift in computation, where people live, work, and play in

Stefan G. Weber
CASED,
Darmstadt, Germany
e-mail: stefan.weber@cased.de

Max Mühlhäuser
Telecooperation Group,
TU Darmstadt, Germany
e-mail: max@informatik.tu-darmstadt.de

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 207–233.
springerlink.com

a seamlessly interweaving computing environment" [56], researchers around the globe try to devise technological solutions that are highly distributed, but still connected, intuitively and non-obtrusively usable, and that aim at preserving data security and privacy while making use of information only when needed and appropriate. While all these issues along with mutual dependencies and system integration aspects need to be taken into account, in order to implement efficient and functional systems, to achieve acceptance among its users, and finally, to obtain commercial success [37], in this chapter, we approach this topic from two specific perspectives:

- in the motivating application context of mission-critical applications, i.e smart collaborative systems that support emergency management work, and
- with a conceptual and technical focus on multilateral security, accountability and data privacy issues.

While true UbiComp requires, on technical levels, efficient and *scalable connectivity* of a multitude of devices and network nodes [2] and means for *intuitive human computer interaction* [38], we believe that, on organizational, legal and social levels, also some support for a *seamless collaboration* of a multitude of users and parties is necessary. However, we stress that multi-user and multi-party applications tend to be also multi-interest applications. Especially, such collaborative applications often highlight inherent conflicts w.r.t. underlying data security and privacy requirements, that are due to different organizational, legal and personal backgrounds of the participants. Obviously, there is no simple answer on how to deal with such inherent security conflicts. The fundamental research question, that our work addresses, is how and to which extend it is possible to realize multilaterally secure [43], i.e. balanced and thus trustworthy, collaborations supported by modern ICT. In this chaper, we focus on collaborations between individuals with strong privacy preferences and organizations with legal documentation and accountability requirements.

1.1 Perspective: Mission-Critical Applications

As forerunners of UbiComp applications, a large range of applications has been proposed during the last years which benefit from fine-grained user tracking in multiple ways. Consider the example of a smart emergency management application: once mobile units are continuously tracked, e.g. by sending their GPS positions and further data to a headquarter, a precise coordination of rescue missions is possible [53] and vital signs of first responders can be monitored. Moreover, units can act as mobile sensors to monitor levels of air pollution, also mission auditing can be supported by creating digital mission logs based on the collected GPS traces [52].

1.2 Inherent Tradeoff: Privacy versus Accountability

While, firsthand, the application scenario strongly indicates benefits of fine-grained user tracking, there is also a further aspect that needs to be taken into account: once it comes to the collection of data related to individuals, organizational and legal documentation and accountability requirements can get in conflict with individual privacy preferences [7]. In the emergency work example, the continuous tracking can also create digital mission logs that document real-world events and that can be analyzed afterwards - even for the detection of inappropriate real-world behaviour. Thus, a mission log could contain information that help to answer questions like ”Did an entity act beyond her competences and authorisations and exploit the current situation with inappropriate, suspicious or even malicious purpose?”. This is a highly critical issue, since, in real-world rescue missions, first responders actually need to break laws in some cases in order to save lives, and the underlying decisions often have to be made under time pressure. Therefore, the psychological burdens of possibly having to face legal consequences due to being digitally accountable need to be addressed, in order to foster acceptance for the use of tracking technologies.

More generally, aspects related to privacy preservation and resulting acceptance issues have been identified as one of the greatest barrier to the success of upcoming UbiComp applications that inherently rely on continuous large scale data collection, like location or employee tracking [44, 9, 49]. We believe that it is a major challenge to design data security mechanisms that allow to reconcile the conflicting goals of accountability for legal and organizational reasons and individual privacy protection [55, 54, 52].

1.3 Approach and Contributions

In this chapter, we describe a new approach for implementing multilaterally secure auditing functionalities for Ubiquitous Computing scenarios. Our work is motivated by and applied to collaborative mission-critical applications, hereby realizing multilaterally secure mission logs, i.e. audit logs for real-world events with special security requirements. Conceptually, the approach consists of two main parts, dealing with data collection and data access issues. The first part, a cryptographic pseudonym construction, allows to build up audit logs, which contain pseudonymized tracking information.

Especially, it enables a mobile user within a location tracking system:

- to protect her location privacy using short-lived transaction pseudonyms that can be created on a personal mobile device, and
- to selectively access audit log content that relates to herself, for own sake.

Second, our work comprises mechanisms for audit log analysis, supporting

- internal audit officers (within an organization) to iteratively analyze the location data logs in a privacy-respecting manner, in case suspecting facts have been reported,
- a lawful disclosure functionality, i.e. an attorney authority may globally revoke privacy protection of individual users, once convincing evidences have been identified.

We stress that, additionally, auditability of the log analysis process is given. Especially, the non-compliance of auditors w.r.t. actions allowed in the log analysis can be detected and reacted upon. Thus, in our work, the principle of *accountability by auditability* is prominent on two conceptual layers, first regarding real world-actions of mobile users as well as regarding the use of auditing functionalities itself. The approach is designed to fit into the framework of context-aware Ubiquitous Computing applications, i.e. it allows for alert generation to support context awareness [27] *on the content level*, while the protection mechanisms are tailored *on the identity level*. Variants of our approach can easily be integrated into existing location tracking infrastructures.

Technically, our work combines pseudo-random number generators, which are used to generate and authenticate transaction pseudonyms, with efficient methods from the area of secure multiparty computation to enable privacy-respecting yet verifiable log analysis. This conceptual combination balances the conflicting security requirements of privacy and accountability in location tracking application to a high extent, hereby implementing a what we claim to be multilaterally secure approach.

1.4 Organization of the Chapter

In the following section, we introduce the background and formulate requirements for multilaterally secure auditing functionalities. Then, in section 3, we sketch an approach to fulfill these requirements. This is followed by a description of necessary building blocks in section 4. Section 5 introduces the main tools and concepts for pseudonym generation and log analysis. This is followed by a presentation of the complete approach in section 6. A discussion and evaluation of our proposal with regards to the security requirements fulfilled and practical aspects can be found in section 7, followed by a discussion of related work in section 8. Finally, in section 9, we sum up and conclude our work.

2 Background and Requirements

In this section, we describe the paradigm of multilateral security, reflect it to our application contexts and derive requirements for multilaterally secure auditing functionalities.

2.1 Multilateral Security and the Role of Accountability

In computer-supported transactions, security requirements of involved parties are often contradicting. Multilaterally secure systems take into account security requirements of all involved parties and aim at balancing contrary interests in an acceptable way [43].

Consequently, after conflicting security requirements have been traded against a multilaterally accepted compromise, the parties should have an incentive not to cheat and only need to minimally trust in the honesty of others, which is, more generally, an ultimate goal in designing security protocols and systems [17]. Rather, the parties can concentrate on reaching a common goal. However, apart from this academic point of view, the actual implementation of multilaterally secure system is a difficult task and active area of research, especially concerning highly distributed and dynamic systems [41].

A basic fact often exploited in the design of multilaterally secure systems and used to enforce correct behaviour is that already the detectability of inappropriate actions and accountability for origination suffices to prevent misbehaviour from happening. In the context of this work, this is what we call the *accountability by auditability* principle.

Traditional technical means to deal with this issue are audit logs [46, 51]. Basically, in an IT system, an audit log contains tamper evident entries that aim at recording irrefutable evidences of all users' actions. While the log content helps to detect inappropriate actions, also users behaving appropriately could use it to defend themselves against false accusations.

2.2 Examples within the Application Scenario

We return to our application scenario of the smart emergency management system. In this system, mobile units and first responders are continuously tracked for the duration of their missions. While the *current* tracking information supports the coordination of rescue missions in the headquarter, collected *historical* position information also creates a log which documents the rescue missions, and thus documents real-world actions. This kind of audit log is what we call a *mission log* throughout this chapter.

The mission log can be analyzed for several purposes in the postprocessing phase of an emergency. In the following, we assume a mission log to obey to a simple structure: it contains several entries in the form *entity ID - time - location*. The organization which accounts for the emergency management wants to be able to analyze processes after a mission, and also, the goal of mission logs is to be able to assign responsibility for real-world actions, since organizations tend to verify compliance. We next describe two motivating real-world use cases of a mission log.

2.2.1 Example: Emergency Car Driving

During the course of a rescue mission, ambulance vehicles or rescue vans sometimes are in need of breaking traffic rules, such as disregarding traffic lights. This may lead to road accidents or injured pedestrians. Usually, emergency cars beckon their emergency missions with sirens, however, there may be situations where no acoustic signals is available.

2.2.2 Example: Cases of Omitted Assistance

Several rescue scenarios involve mobile first responders. While such forces are on their mission, they strongly priorize actions, according to given instructions. However, sometimes they depart from that. Also, when it comes to dealing with injured persons, rescue forces act according to triage regulations, i.e. according to the severity of injuries they postpone or even skip treatment, which might be considered by eye witnesses as cases of omitted assistance. Moreover, there are a lot of situations that entail the destruction of properties, like breaking doors to enter a building, that are relevant to after mission warranty and accountability discussions.

2.3 Towards Balanced Auditing Functionalities

Having introduced the concept of the mission log, as a special kind of audit log for real-world actions during emergency missions, we next discuss and derive basic security requirements that have to be met to implement a *multilaterally secure mission log*. Thereby, multilateral security is considered in the sense that we want to take into account security and privacy requirements of mobile users, emergency management organizations and law enforcement agencies, as well.

First, allowing the mobile units to be tracked pseudonymously[1] instead of by means of a fixed ID implements a very basic kind of privacy protection in our application. This is necessary to address the inherent privacy issue in the location tracking. Consequently, following this approach also requires auditing functionalities that are compatible with pseudonymized log data. While this is the starting point for designing the system, however, in order to achieve a multilaterally secure solution, a more complex security design is required. According to [21], two important facts have to be dealt with in the context of creating privacy-respecting log analysis functionalities:

1. ”The controlled disclosure of pseudonyms is the controlled ability to make pseudonymized objects accountable again. This ability is controlled by controlling who can use the ID to pseudonym mapping.”
2. ”The disclosure of pseudonyms should be bound to a priori specified purposes.”

Thus, a central point is to implement a selective control functionality regarding the pseudonym linkability for users, organizations and authorities. Moreover, mechanisms that allow for the detection of suspicious evidences inside such a log are required. According to our previous discussion, we propose the following set of functionalities in order to reconcile privacy and accountability as fair and as far as possible.

[1] Historically, the term *pseudonymous* relates to the Greek word *pseudonymos*, which means *having a false name* [19]. In a technical sense, a pseudonym is an identifier of an entity that is used instead of the entity’s real-world name [42]. Pseudonymity is the use of pseudonyms as identifiers.

First, regarding the perspective of the user:

- The user should be tracked under short-lived transaction pseudonyms [42] to provide some kind of basic location privacy protection.
- The user should be able to authenticate a pseudonym, that he has been tracked under. In dispute cases, this functionality allows her to access the mission log in order to repudiate false accusations by providing evidences of exoneration.

Second, regarding the perspective and role of third parties, e.g. emergency management organizations[2]:

- It should be possible to selectively analyze the entries that are recorded in a log in a privacy-respecting manner. Therefore, the logs should contain only pseudonymized entries that hide the real-world identities of tracked persons.
- It should be possible to link sample entries, i.e. to verify if they belong to the same user.
- It is desirable to check if one entry relates to a common organizational structure or function.
- If legally convincing pieces of evidence for misbehavior have been identified in the course of this analysis, it should be possible to reveal the true identity of a tracked person.
- An operational separation of duty should be enforced, i.e. no single entity should be able to (mis)use the audit functionalities.
- The whole process of log analysis should also be auditable, i.e. it should be detectable if the parties that are responsible for it do not comply with the rules set up for the privacy protection of the individual user.

Third, the legal perspective needs to be taken into account:

- It should be possible to exercise a global lawful disclosure functionality, i.e. an attorney authority may revoke privacy protection of any users in question, once convincing evidences of inappropriate behaviour have been identified. This is a requirement for many ICT applications, that actually may lead to court proceedings for accused offences.

3 Construction Idea

As discussed in section 2.3, pseudonymous auditing can balance the conflict between accountability and privacy to a certain extent. From a different point of view, a pseudonym implements the central reference point to evidences in an audit log.

It is known that techniques from the area of secure multiparty computation (SMPC) can theoretically be applied to a large range of problems in the area of privacy-preserving data analysis [30, 35]. Basically, SMPC [57] allows for implementing privacy-respecting multiparty protocols that do not rely on a single trusted third party

[2] The parties that actually exercise the log analysis functionalities for the organizations are called *audit officers* throughout this chapter.

(TTP). The intention of these cryptographic techniques is that a number of distinct, but connected parties may jointly compute an agreed function of their inputs in a secure way. Hereby, the correctness of the output as well as the privacy of each input shall be preserved, even if some participants cheat[3]. Generally, secure multiparty computation is an approach to distribute the functionality and powers of a single TTP among several parties. However, this is traditionally only achieved with very high computational costs [35], due to the intensive use of secret sharing [47] and operations on secret shared data in SMPC protocols. However, more efficient special purpose approaches to SMPC have been proposed, e.g. in the *mix-and-match* approach [31], secret sharing techniques are replaced by operations on encrypted data.

Our approach follows the spirit of the *mix-and-match* approach. Thus, we formulate the pseudonym generation in terms of specific encryption operations. This enables us to apply efficient concepts from SMPC for realizing privacy-respecting log analysis functionalities *on the pseudonym level*. Additionally, we make use of seeded PRNGs to control random factors inside the encryption operations. This allows us to implement a further direct identity-to-pseudonym-mapping, that can be exploited by individual users, in order to access log entries that relate to herself.

4 Building Blocks

In this section, we briefly describe the main building blocks that are employed in our approach.

4.1 Cryptographically Secure PRNGs

A pseudo-random number generator (PRNG) [34] is a deterministic algorithm for generating sequences of numbers. Hereby, the created numbers apparently exhibit the properties of real random numbers. PRNGs incorporate an internal source of entropy, which is called a seed, to derive and compute their output. Cryptographically secure PRNGs are special kinds of PRNGs that produce sequences of numbers with strong security requirements, i.e. it is actually impossible to guess or derive any forward or backward numbers by analyzing the output of such a PRNG. Our constructions actually employ cryptographically secure PRNGs. However, in the following, we often simply refer to them as PRNGs.

4.2 Threshold ElGamal Cryptosystem

A key primitive in our approach is the ElGamal cryptosystem [16], over subgroups $\mathbb{G}_q$ of order q of the multiplicative group $\mathbb{Z}_p^*$, for large primes $p = 2q + 1$. We treat the primes p, q and a generator g of $\mathbb{G}_q$ as common system parameters. ElGa-

[3] A classic example of SMPC is the millionaires' problem due to Yao [57]: some millionaires (*the parties*) want to find out, who is the richest (*agreed function*) without revealing the precise amount of their individual wealth (*input privacy*).

mal encryption is know to be semantically secure in $\mathbb{G}_q$, under certain complexity assumptions [50]. Practically, semantic security means that no partial information about a plaintext is leaking from the corresponding ciphertext. More specifically, we utilize a threshold variant of the ElGamal cryptosystem, according to Cramer et al. [13], which allows to distribute cryptographic operations and thus offers robustness and distributed trust. In the system, a private key $s \in_R \mathbb{Z}_q$ can be defined in two ways: first, it can be initially generated by a trusted dealer and then be secret shared [47] among all n participating authorities. Second, it can be generated via the distributed key generation protocol of Pedersen [40], whereby no single party knows the complete key. In both ways, w.r.t. the participating authorities, the power to decrypt is distributed among all of them, and a quorum, i.e. a minimal majority of t out n authorities need to cooperate to perform a threshold decryption. The authorities common public key is $h = g^s \bmod p$. A message $m \in \mathbb{G}_q$[4] is non-deterministically encrypted by choosing $r \in_R \mathbb{Z}_q$ and by computing $(g^r, h^r m)$.

In our work, the communication model in the threshold cryptosystem consists of a broadcast channel with memory. This channel is used to store and exchange information in any protocol involving distributed computations, e.g. partial decryptions and identifiers of each authority in a threshold decryption. Moreover, in our setting, the broadcast channel is append-only, i.e. once the information is sent over it, it is stored and cannot be changed or deleted afterwards. Thus, its content can later be analyzed for audit purposes.

4.3 Non-interactive Zero Knowledge Proofs

Zero knowledge proofs (ZKPs) [24], which are basically generalized challenge-response authentication protocols, are used to guarantee correctness of and participation in distributed cryptographic operations and protocols. Especially, we use non-interactive zero knowledge proofs (NIZKPs) by applying the Fiat-Shamir heuristic [18]. See e.g. [12] for discussions of these techniques. NIZKPs can be stored, and may be verified, comparable to digital signatures, after a protocol run. Thus, incorrect actions can be deduced by looking at those digital transcripts. NIZKPs are a common cryptographic approach to implement auditability[5].

4.4 Plaintext Equality Tests

A plaintext equality tests (PET) [31] is a primitive for pairwise blind comparison of ciphertexts of non-deterministic threshold cryptosystems like ElGamal. A PET allows to test whether two ciphertexts represent the same plaintext by performing algebraic operations on the ciphertext, but without revealing the plaintext. PETs and generalizations thereof are key primitives to implement privacy-respecting log analysis functionalities. The use is described in detail in section 5.3.

[4] Messages that are not in $\mathbb{G}_q$ can efficiently be mapped onto $\mathbb{G}_q$ [31]. Thus, arbitrary strings can be encrypted.

[5] In research dealing with cryptographic protocols, this is often referred to as *verifiability*.

4.5 *Reencryption Mixnets*

A mixnet, originally introduced by Chaum [10], is a cryptographic tool to anonymize sets of ciphertexts. In our work, we employ ElGamal based reencryption mixnets [39], which basically reencrypt and permute ciphertexts in order to anonymize them. In this setting, reencryption can be done without a private key. Moreover, we also require the mixnets to be verifiable, i.e. to provide NIZKPs of correctness of their operations. We do not go into details here, but point to the proposal of Furukawa et al. [23], that implements such cryptographic auditability for reencryption mixnets. We apply the underlying concept of reencryption in two stages of our work, first to derive transaction pseudonyms (cf. section 5.1), second, to build up anonymous reference sets used in re-identification steps of our scheme (cf. section 5.3).

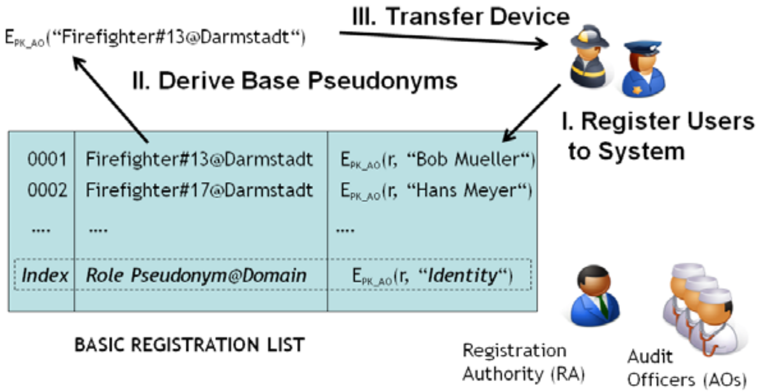
5 Basic Tools and Concepts

We now introduce the basic constructions and protocols of our approach. These constructions implement functionalities for pseudonym mapping and linking, for all considered parties, i.e. users, organizations and law enforcement agencies.

5.1 *Registration and Generation of Pseudonyms*

In this section, we introduce the concepts for pseudonym generation. Basically, we propose to encode a static attribute inside malleable pseudonyms, by generating pseudonyms as encryptions of an attribute under the public key of a threshold ElGamal cryptosystem. The resulting construction is what we call a *pseudonym with implicit attribute*.

First, each user must participate in a registration phase, which is depicted in Figure 1. In this phase, each user receives a trusted personal device which includes


E_{PK_AO}("Firefighter#13@Darmstadt")
III. Transfer Device
II. Derive Base Pseudonyms
I. Register Users to System
BASIC REGISTRATION LIST
Registration Authority (RA)
Audit Officers (AOs)

0001	Firefighter#13@Darmstadt	E_{PK_AO}(r, "Bob Mueller")
0002	Firefighter#17@Darmstadt	E_{PK_AO}(r, "Hans Meyer")
....	....	....
Index	*Role Pseudonym@Domain*	E_{PK_AO}(r, "*Identity*")

Fig. 1 Registration Phase

a cryptographically secure PRNG. Therefore, the user interacts with a trusted registration authority (RA). The registration phase consists of the following main steps:

I. Each user is added to an integrity protected registration list. He receives a distinct role pseudonym, which associates the user with the issuing organization. For example, a user can be registered as *Firefighter*#13@*Darmstadt*. The user's real-world identity, e.g. *Bob Mueller*, is encrypted and stored together with the role pseudonym on the registration list.
II. The RA derives the base pseudonym by encrypting the unique role pseudonym. Thus, each base pseudonym encodes a distinct implicit attribute.
III.The RA uploads the base pseudonym to a personal device. The user receives this personalized device.
IV.The user generates and registers a unique seed in the PRNG of the device to enable it for pseudonym generation.

In the registration phase, the encryptions are done under the public key belonging to a set of so called audit officers[6] inside the organization, e.g. in the emergency management organization:

- The real-world identity ID is encrypted as: $E_{PK_{AO}}(ID) = (g^r, h^r ID)$.
- To generate the base pseudonym for a user, the registration authority encrypts the chosen attribute A as: $P = E_{PK_{AO}}(A) = (g^{r_s}, h^{r_s} A)$. Moreover, the random value r_s, the start value for pseudonym generation, is also transferred to the user and stored on her device.

The user is now able to derive *transaction pseudonyms* from her base pseudonym in the following way:

1. The seeded PRNG is used to generate a sequence of random numbers.
2. Each random number r_i is used to compute a randomization factor $F_{r_i} = (g^{r_i}, h^{r_i})$.
3. F_{r_1} is used to construct the *first* transaction pseudonym by multiplying it with the base pseudonym: $P_1 = (g^r, h^r A) * (g^{r_1}, h^{r_1}) = (g^{r+r_1}, h^{r+r_1} A)$.
4. Further transaction pseudonyms are created by repeated multiplications: $P_{i+1} = P_i * F_{r_{i+1}}$.

By this procedure, a user creates a set of different transaction pseudonyms that all contain the same implicit attribute. These pseudonyms are used instead of fixed IDs during the location tracking.

5.2 Log Attestation

Due to the construction, presented in the last section, users are also enabled to authenticate a transaction pseudonym that is stored in a mission log. Therefore, a

[6] Cryptographic keys have been generated in a previous setup phase, cf. section 6.1. The audit officers *share* the private key belonging to the single public key. Due to the use of threshold cryptography, variants with different distributions of power are possible, cf. section 7.

user needs to show that she is in possession of the base pseudonym and the correct aggregated random factor, which allows to reproduce a recorded pseudonym, thereby authenticating a complete log entry. This functionality is what we call *log attestation*.

5.3 Log Analysis

In this section, we introduce the concepts for privacy-respecting mission log analysis. Basically, we harness the possibility to do algebraic operations on the non-deterministic encrypted ciphertexts that represent pseudonyms.

Remember the structure of a mission log, as introduced in section 2.2: *entity ID - time - location*. Assuming that the *entity ID* values are actually values of *pseudonyms with implicit attributes*, we allow the audit officers of the responsible organization to execute *two basic operations* for privacy-respecting log analysis:

1. check if two log entries relate to the same entity but without revealing the actual ID of the entity;
2. check if one entry relates to a group of entities with a common organizational role or function.

The first operation is implemented by executing a *plaintext equality test* on the pseudonym values of two log entries. Suppose that $P_a = (g^{r_a}, h^{r_a}A_a)$ and $P_b = (g^{r_b}, h^{r_b}A_b)$ represent two entries of that kind. If they relate to the same entity, they contain the same implicit attribute. In order to verify this, the pseudonyms can be algebraically divided: $P_c = P_a/P_b = (g^{r_a - r_b}, h^{r_a - r_b}A_a/A_b)$. Given that A_a equals A_b, this is an encryption of the attribute ”1”. By performing a threshold decryption, the audit officers yield an explicit attribute which is either ”1” or a meaningless different value.

The second operation is an extension of the procedure above to a global instead of pairwise comparison of pseudonyms. It allows to test if one log entry relates to a certain organizational unit, function or even place. For example, it can be tested whether the entry relates to ”Response Team Alpha” or ”fire department Darmstadt”. Especially, this test does not disclose which of the possibly involved entities is the actual originator of the log entry.

The method of the *generalized blind plaintext equality test* is introduced next. Again, it is based on operations of an ElGamal threshold cryptosystem. Basically, it works as following:

1. First, all participating audit officers jointly generate a shared key z.
2. Then, the authorities select all attribute values that are relevant to the organizational function or role and create base pseudonyms for each relevant attribute.

3. Next, all base pseudonyms are processed by a reencryption mixnet. This creates an anonymized list of base pseudonyms, i.e. the positions of the individual attributes in the list as well as ciphertext representations are changed.
4. The audit officers cooperatively apply their shares of z to each anonymized base pseudonym. This process achieves blinding of the attribute inside the pseudonyms.
5. After that, each blinded pseudonym is jointly decrypted. This yields a blinded attribute, which is used as a deterministic yet blind fingerprint of the original attribute related to a relevant base pseudonym.
6. After processing all base pseudonyms that need to be considered, they can be compared without leaking information about the implicit attribute by comparing only the blind fingerprints.
7. In order to do so, the authorities also derive a blind attribute fingerprint for the pseudonym value of a log entry that is to be verified.

Having outlined the abstract steps of the method, we next go into more details. The whole scheme makes use of secret sharing techniques according to Shamir [47] and of the distributed key generation according to Pedersen [40]. First, to jointly generate the secret shared key z used for blinding, the audit officers employ the distributed key generation protocol due to Pedersen. In this protocol, each audit officer AO_j receives a share z_j of the key z. Also, each officer is publicly committed to the share z_j by a public value $\rho_{z_j} = g^{z_j}$, due to the execution of the protocol.

In the following, we describe the complete protocol for *distributed blinding*, which is analog to the distributed decryption protocol [13] of the ElGamal threshold cryptosystem. This protocol can be used to blind an arbitrary element $x \in \mathbb{G}_q$ using the shared key z. The following method is used to apply z cooperatively to the pseudonyms[7]:

1. Each officer computes $b_j = x^{z_j}$, a partial blinding of x, by applying its secret z_j. Also, each officer publishes b_j together with a NIZKP that

$$\log_g \rho_{z_j} = \log_x b_j$$

   The latter is realized using a proof of knowledge for equality of discrete logs [11]. The proof assures that the officer indeed utilized the correct share to produce the partial blinding[8].
2. For any subset Λ of t authorities with valid zero-knowledge proofs, the complete blinded value x^z is reconstructed using the discrete Lagrange interpolation

$$x^z = \prod_{j \in \Lambda} b_j^{\lambda_{j,\Lambda}} \bmod p$$

[7] Pseudonyms are effectively encoded as two elements of $\mathbb{G}_q$, whereas the second element is directly derived from the implicit attribute.

[8] Otherwise, the use of a fake share would lead to an incorrectly blinded value, which could not be used for blind matching purposes.

where

$$\lambda_{j,\Lambda} = \prod_{l \in \Lambda \setminus \{j\}} \frac{l}{l-j} \bmod q$$

are the appropriate Lagrange coefficients.

Now, let $A_i \in \mathbb{G}_q$ be an attribute plaintext[9], and $(g^r, h^r A_i)$ an pseudonym with implicit attribute A_i with $r \in_R \mathbb{Z}_q$, the authorities produce the deterministic fingerprint through the following steps:

1. To each component of $(g^r, h^r A_i)$ the distributed blinding protocol is applied, blinding it to a fix secret shared exponent $z \in \mathbb{Z}_q$: $((g^r)^z, (h^r A_i)^z) = (g^{rz}, h^{rz} A_i^z)$.
2. The blinded pseudonym is jointly decrypted to the blinded attribute A_i^z using the distributed decryption protocol of the threshold ElGamal cryptosystem.

Now, A_i^z represents a deterministic fingerprint produced with a key z. It is used to blindly compare attributes encoded in the pseudonyms.

5.4 Disclosure of Pseudonyms

As stated earlier, ”the disclosure of pseudonyms should be bound to a priori specified purposes” [21]. In the last sections, we provided methods for analyzing entries of a mission log with regard to linkability of pseudonyms to chosen attributes. Rules and conditions for the actual detection of inappropriate behaviour and misuse must be defined in the organizational and legal context of the application, according to which the audit officers have to decide on whether detected patterns inside the log provide convincing evidences. However, we propose to design the rules according to the need-to-know principle [3], i.e. to do a step-wise analysis of the log with the highest possible anonymity restrictions.

In order to complement the two operations for log analysis, we describe the operation for the complete disclosure of pseudonyms next:

- Upon misuse detection, the audit officers decide to completely disclose the identity associated with a pseudonym.
- First, they cooperatively decrypt the pseudonym part of the relevant entry of the mission log. This yields the plaintext of the distinct attribute encoded in the pseudonym.
- Next, the audit officers select the corresponding entry on the registration list. Then, they cooperatively decrypt the deposited ciphertext to reveal the real-world identity.

The lawful disclosure functionality works in a comparable way. Basically, we assume that the global attorney also plays the role of the trusted dealer in the key generation process (cf. section 4.2), thus this authority is in possession of the private key of the audit officers.

[9] We simplify the description at that point, $A_i \in \mathbb{G}_q$ is actually only the algebraic representation of an attribute. Mapped back to a string representation, it could be ”Firefighter #13@Darmstadt”.

- First, the global attorney decrypts the pseudonym part of the relevant entry of the mission log. This yields the plaintext of the distinct attribute encoded in the pseudonym.
- Second, the authority selects the corresponding entry on the registration list. Then, it decrypts the deposited ciphertext in order to disclose the real-world identity.

6 Complete Approach

This section presents our complete approach. First, we sum up the phases. Then, we depict the overall scheme in more detail.

6.1 Phases

- *Setup Phase:* In this phase, system parameters for the threshold cryptosystem and cryptographic keys used for pseudonym generation and auditing are created.
- *Registration phase:* Each user receives a personal device. Also, a base pseudonym with an implicit attribute is registered to each user. The user locally creates and registers a unique seed in the PRNG of the device. The base pseudonym, together with the seeded PRNG, allows to derive transaction pseudonyms in the next phase.
- *Pseudonym Generation Phase:* A user creates transaction pseudonyms locally on the personal device. Each transaction pseudonym is a non-deterministic threshold encryption of the same implicit attribute. The devices PRNG provides random factors used in the pseudonym creation.
- *Tracking Phase:* During a mission, the user is regularly tracked under transaction pseudonyms. The pseudonym in use is regularly changed, according to the preference of the user.
- *Log Analysis Phase:* In this phase, entries of the log are processed by audit officers for organizational and legal reasons. Upon convincing detection of inappropriate actions, pseudonyms can be disclosed.
- *Log Attestation phase:* After the mission, a user may authenticate single entries of the mission log by providing the correct keying material that allows to reconstruct the pseudonyms of the log entries.
- *Officer Auditing Phase:* In this phase, the appropriateness and correctness of the actions of the audit officers in the log analysis phase is verified. This done by checking the NIZKP stored on the broadcast channels with memory,
- *Lawful Disclosure Phase:* This phase is optional. If a mission log analysis leads to court proceedings, a global attorney can re-identify any log entries that may support returning a verdict.

6.2 Scheme Description

We next describe our complete approach in detail. Thereby, we take into account the building blocks from section 4, and the operations described in section 5.

In the description, we denote the participants as: U_i a user of the location tracking system; AOs the n audit officers that share the private key SK_{AO} with the corresponding public key PK_{AO}; RA a trustworthy registration authority; GA a global attorney authority.

- *Setup:* The GA acts as trusted dealer, she selects system parameters p, q, g and generates PK_{AO}, and SK_{AO}, appropriate to the threshold ElGamal cryptosystem. PK_{AO} as well as the ElGamal system parameters are published. Once the participating audit officers (AOs) have been appointed[10], they receive individual shares of SK_{AO}. In addition, the AOs produce a shared key z.
- *Registration:* Each user U_i is registered by a trustworthy registration authority RA. First, a mobile device is selected for personalization. Then, each user receives a base pseudonym with an implicit attribute: $P_{U_i,B} = E_{PK_{AO}}(A_{U_i})$. The random factor $r_{i,B}$ used in the encryption process is stored on the user's device. The device is transferred to the user. The user generates an unique seed s_{U_i} and registers it to the PRNG of her mobile device. At the end of this phase the RA publishes a reference list C. It contains entries of all registered users (encrypted real world ID) and their assigned attributes: $C_{U_i} = E_{PK_{AO}}(U_i) - A_{U_i}$.
- *Pseudonym Generation:* Each user U_i derives from her base pseudonym $P_{U_i,B}$ a set of transaction pseudonyms $\{P_{U_i,j}\}$. In order to do so, first, the seeded PRNG is used to produce a set of random factors $\{r_{i,j}\}$. Then, the random factor inside the base pseudonym is updated with a random factor from the set: $P_{U_i,1} = P_{U_i,B} * (g^{r_{i,1}}, h^{r_{i,1}})$, $P_{U_{i,j+1}} = P_{U_i,j} * (g^{r_{i,j+1}}, h^{r_{i,j+1}})$.
- *Tracking:* When using the tracking system, each user is tracked under transaction pseudonyms from the set $\{P_{U_i,j}\}$. According to the preferences of each user, the pseudonym $P_{U_i,j}$ is changed to $P_{U_i,j+1}$ in a specified interval of time. Note that changing a transaction pseudonym does not change the implicit attribute an user is tracked under. In the tracking phase, a mission log is created. Its entries are in the form *entity ID - time - location*. The *entity ID* field records the value of a pseudonym with implicit attribute, i.e. a transaction pseudonym.
- *Log Analysis:* According to an organizationally and legally defined set of conditions and rules, the audit officers use the provided operations to detect evidences of misuse in the mission log. If convincing evidences have been identified, they induce a pseudonym disclosure. Therefore, the authorities cooperatively decrypt the pseudonym part of the relevant entry of the mission log. The resulting attribute plaintext is used to select the corresponding entry on the registration list. Then, the authorities cooperatively decrypt the deposited ciphertext to disclose the real world ID.

[10] This can be delayed in time, since the public key, which is needed to generate pseudonyms, has already been created.

- *Log Attestation:* In dispute cases after the tracking phase, a user U_i may access data recorded in the mission log. In order to do, she selects an entry k of the mission log by handing out authentication information in form of an aggregated random factor $r_{i,A} = \sum_{j=1}^{j=k} r_{i,j}$ and the base pseudonym $P_{U_i,B}$ to the audit officers. After verification, whether the pseudonym of the mission log entry k matches the reconstructed transaction pseudonym $P_{U_i,k}$, the user receives a tuple *time - location*, which is additionally certified by the audit officers. The user can use the tuple to repudiate location- and time-dependent accusations against her.
- *Officer Auditing:* Internal and external auditors may verify the appropriateness and correctness of actions of the audit officers in the log analysis phase. In order to do so, they read out the content of the broadcast channel with memory. The channel provides a log of the committed actions. Actions that do no comply with the allowed operations can be identified, as well as attempts to corrupt the cooperative operations, since stored NIZKPs cannot be verified correctly in that case.
- *Lawful Disclosure:* In case a mission log analysis did not clear all dispute cases, court proceedings can be initiated afterwards. Herein, the global attorney *GA* can re-identify any log entries that may support returning a verdict. The *GA* uses SK_{AO} to decrypt the pseudonym parts of any relevant entry of the mission log. Any plaintext, i.e. the implicit attribute, distinctly refers to one entry of the registration list. The *GA* decrypts any relevant entry of the registration list to recover the real-world identity of the originator of the log entry.

7 Example and Discussion

Having introduced the concepts and the complete scheme, in this section we first describe an application example. Then, we discuss the presented approach with respects to its main security requirements. Moreover, we elaborate on practical aspects that have to be considered in the course of implementing a system that actually employs our concepts.

7.1 Application Example

We next provide an example that illustrates how our approach can actually be employed in practice for analyzing mission log data. Hereby, we especially refer to section 5.3 w.r.t. the two basic operations of log analysis. We assume the scenario, that a large airport is affected by a large scale emergency. Several airplanes have caught fire due to an accident. The burning is spreading over to the terminal buildings. Since the fire brigade of the airport is unable to handle the situation on its own, additional forces from nearby fire departments are requested. Also, the fire department Darmstadt sends a group of 50 mobile first responders to support the rescue missions. Arriving at the airport, the first responders register to the tracking system, receiving base pseudonyms with implicit attributes *Firefighter* #247@*Darmstadt* to *Firefighter* #296@*Darmstadt*. During the course of the successful rescue mission, their movements and actions are continuously tracked and stored in the mission log. In the postprocessing phase of the emergency, a group of airport officers is appointed to analyze a specific incident which has been reported to them by an anonymous eye witness: it is mentioned that a group of 4 firefighters suddenly disappeared from an important task to extinguish fire in an office wing, located in the vicinity of a jeweller in the shopping area. The store reports that expensive items have disappeared, and several offices have been destroyed due to the lack of man power. Analyzing the mission log, the officers manage to identify traces of 5 pseudonymous entities that move away from the office wing in question (*by using the first operation*). Moreover, these 5 pseudonymous entities can be identified to belong to *fire department Darmstadt* (*by using the second operation*). On request, the commander from Darmstadt asserts, that a special group of his first responders, that is known as *Response Team Alpha*, decided to change the mission task short-run, due to the observation of strange knocking sounds nearby. The audit officers define the *Response Team Alpha* unit and manage to relate the traces thereto (*by using the second operation*). While, in that case, the commander's information and the information from the audit process suffices to resolve the incident in question, the individual firefighter could have also used the *log attestation* functionality, to document scenes of the mission. After the audit, the firefighters access the transcripts of the audit officers' actions. They can recognize their appropriate behaviour during the log analysis.

7.2 Security Properties

- *Attacker Model:* Our work is specified as a multilaterally secure approach, i.e. firsthand, the involved parties are seen as mutual attackers. However, the intention of the approach is to provide an *enabling technology* that allows to realize an acceptable compromise between individual privacy and accountability. The protection mechanisms are devised on the application level, i.e. data that is leaking on lower layers is not explicitly considered and could be exploited by further attackers. However, the approach is tailored within the context of emergency communications where parties use dedicated digital radio networks with own security infrastructure, e.g. according to the TETRA standard[11]. Thus, complementary security measures are provided by the host network.
- *User Privacy and Pseudonym Linkability:* From the users' perspective, individual privacy within the location tracking application is protected due to the use of transaction pseudonyms. In fact, any user is able to adjust the frequency of pseudonym changes and also of location updates sent to the headquarter, thus she is empowered to adjust the level of linkability within a mission log. Generally, concerning the provided degree of linkability, pseudonyms can be classified into *transaction pseudonyms*, *role pseudonyms*, *role-relationship pseudonyms* and *person pseudonyms* [42]. Our approach can be seen as a conceptual combination of transaction and role pseudonyms. The ability to use short-lived random pseudonyms in a freely determined frequency relates to the first property, while

[11] Cf. WWW.TETRAMOU.COM

the implicit attribute can be interpreted as a role pseudonym that can be read out in several levels of granularity, e.g. ”Firefighter”, ”Response Team Alpha”, ”Firefighter #13@Darmstadt”.

- *Privacy-Respecting Log Analysis:* The basic pseudonym construction stems from a semantically secure encryption operation, thus pseudonyms by itself do not leak information about the implicit attribute. Moreover, no single audit officer is able to decrypt a pseudonym and thus to link a pseudonym to a user. This is achieved since both registration list and pseudonym creation are performed as encryption operations of a threshold cryptosystem. Therefore, audit officers have to cooperate in any step of the log analysis. This implements an operational separation of duty in our system. During the course of the analysis, the officers are able to build up reference sets of pseudonyms with implicit attributes, that they test log entries against. Since anonymity is defined according to the size of the anonymity set [42], interestingly, this method allows to obey regulations for log analysis that are formulated in terms of anonymity.
- *Pseudonym Authentication:* In order to implement a log attestation functionality, which aims at defending against false accusations and thus could actually entail legal effects, the pseudonym authentication needs to assure a uniqueness property. Especially, it must not be possible for an attacker[12] to provide authentication information for a pseudonym without being the real originator. This is achieved in our construction due to a two-factor authentication process: The users needs to provide both the base pseudonym and the aggregated random factor. An attacker could only guess both base pseudonym and matching aggregated random factor, but he cannot access a seed to derive correct values, since the seed is generated locally and the registration is assumed to be trustworthy. Moreover, the probability to succeed with such a bruteforce attack can be lowered by requiring a user to authenticate multiple entries in a log attestation. From a broader perspective, the final acceptance of tracking information as evidences in proceedings is a legal issue. Especially, in a trial, it has to be considered if evidences of *having been at a certain location* actually relate to *having committed a certain action*. From a security point of view, this boils down to the question, whether the location tracking infrastructure is trustworthy, which is, in fact, a core assumption of our approach.
- *Distribution of Powers:* The proposed construction allows to flexibly represent a large range of legal and organizational distributions of duties and powers. This is due to the use of a (t,n) threshold ElGamal cryptosystem as well as the distributed computation of fingerprints. Firsthand, the approach can also tolerate the failure of at maximum $n-t$ authorities, which may be due to unavailability or due to corruption. Additionally, the robustness of scheme stems from its ability to tolerate attacks against audit officers or failures of them, without corrupting the whole system. However, we require the registration phase of the scheme to be trustworthy, since it depends on a single registration authority *RA*, which is contrary to the distributed design of the rest of the scheme. Also, the global

[12] In that case, the attacker could be a further user of the location tracking system who tries to fake evidences.

attorney *GA* authority represents such a single TTP. However, the global attorney can also be avoided, if not required in a certain application context. In that case, the initial generation of SK_{AO} is executed according to the distributed key generation protocol of Pedersen [40]. Also, the power of the *AOs* to completely disclose pseudonyms can be avoided, by encrypting the registration list under PK_{GA} instead of PK_{AO}. Variants in the encryption of the registration list can also be used to impose higher computational cost on the final disclosure step, if necessary.

- *Accountability by Auditability:* The multilateral security property of the overall approach relies on the principle of *accountability by auditability*: mobile units can be held accountable for real-world actions during their missions, due to the existence of mission logs. Audit officers can be held accountable for their actions during the log analysis process, due to existence of transcripts of the distributed cryptographic operations. Especially, anyone who can read the broadcast channel can audit the log analysis process. The existence and correctness of each distributed computation step can be verified by checking the broadcast NIZKPs stored on the broadcast channel. However, due to the use of zero knowledge techniques, this does not leak additional information. Thus, only the correctness of the scheme and appropriateness of the process are made transparent to external parties.

7.3 Practical Aspects

- *Handling short-lived Identifiers:* In order to use transaction pseudonyms in the location tracking, we assume that the personal device of the user is able to handle changing identifiers. Especially, we assume that the tracking does not proceed under a static MAC address. The technical feasibility to implement such changing identifiers is a standard research assumption, we share this concept e.g. with [26, 28, 25]. Moreover, this approach requires the availability of special mobile devices. This is an organizational fact that is actually true in the emergency management context, where any user is handed out e.g. a personalized TETRA device before the missions start. Additionally, a variant of our approach can easily be integrated into existing location tracking infrastructures. In this *synchronized case*, the actual tracking can proceed under a static identifier. Instead, both the mobile devices as well as a trusted pseudonymizing module (for each user) compound to the location tracking system generate the same chain of transaction pseudonyms, in the same frequency, derived from the same seed value. Thus, pseudonymization of the mission log is achieved right after data collection, not within. This approach requires less specialized devices and is conceptually comparable to the *RSA SecurID time-based one time passcodes* [13], where PRNG-based reference values used for authentication are dually generated on user-owned hardware tokens as well as on trusted servers within a company.
- *Pseudonym Bit Size:* In the proposed scheme, pseudonyms are encoded as *two* elements of an underlying algebraic group. Therefore, the actual bit size of a

[13] Cf. HTTP://WWW.RSA.COM/NODE.ASPX?ID=1156.

pseudonym depends on the characteristics and chosen parameters of the group. In order to minimize the communication and storage overhead, it is possible to employ elliptic curve groups in order to implement the cryptographic operations of the ElGamal cryptosystem.

- *Mixnet Operators:* Reencryption mixnets are used within the log analysis in order to anonymize the list of base pseudonyms. We propose to use verifiable and thus auditable mixnets. Their operation is not a time critical issue, however it is an organizational question who actually operates the mixnets, i.e. which party provides the servers for their operation. In our scheme, a trustworthy registration authority is already required. Practically, this party can also operate the mix servers. Hereby, it can also share the broadcast communication channel with the threshold cryptosystem to store NIZKPs.
- *Context Awareness:* Generally, UbiComp applications entail some kind of proactive system behaviour. On a technical level, this partly requires that alerts can be generated, e.g. on acquired sensor data. In the application example of first responder tracking, vital data or air pollution can locally be sensed and also be send to the headquarters. There, the monitoring system can generate alerts if certain thresholds are exceeded, e.g. if physical parameters indicate injuries. Staff member can react upon such warnings. In order to realize such functionalities, *data content* must be efficiently analyzable. Our approach designs protection mechanisms in form of pseudonyms on the identity level, leaving the content level for real-time alert generation.

8 Related Work

In this section, we review and discuss representative related research approaches. Existing research can be classified into work regarding linkable pseudonyms, their applications in location privacy protection as well as in privacy-respecting audit log analysis.

8.1 Linkable Pseudonyms

Historically, Chaum [10] introduced digital pseudonyms as a basic tool for privacy protection in distributed systems, by implementing a firsthand unlinkability between a real-world identity and a pseudonymized identity. In the following years, several types of pseudonyms and applications have been identified, and a wide scope of scientific background has evolved [42]. Concerning the provided degree of linkability, pseudonyms can be classified into *person pseudonyms*, *role-relationship pseudonyms*, *role pseudonyms* and *transaction pseudonyms*, whereby the degree of unlinkability and thus anonymity is highest for transaction pseudonyms. Viewed differently, linkable pseudonyms are pseudonyms that additionally encode secret trapdoor information, to enable attribution of multiple pseudonyms to one or more real-world identities. Different from our work, linkability is usually only possible for either third parties or the user herself [45], not for both. Recent cryptographic research abstracts from pseudonyms and focuses on separating authentication from identification issues [8], but also allows for reconciliation thereof, to construct so-called self-certified pseudonyms [36].

8.2 Location Privacy Protection

In the context of location tracking applications and UbiComp, pseudonyms have been proposed as one of the basic means for *location privacy protection*. Location privacy has been identified as a key factor to personal wellbeing and safety [15] and protection from intrusive inferences. In our work, location privacy is a key to resolve possible real-world accusations. In a traditional direction of research, pseudonyms are used to protect against attackers that try linking several pseudonyms in order to construct movement profiles and comprehensive user traces. In early work, Kesdogan et al. [33] proposed to use multiple short-lived, thus changing pseudonyms in mobile GSM networks. Beresford et al. [4] combined the use of changing pseudonyms with an geographic abstraction of mixnets, to form so called mix-zones. For users of location-based services, a mix-zone is a region without service use, in which the actual pseudonym change is done, to hinder profiling. Recent follow up work [22] addresses non-cooperative location privacy models and evaluates the effect of unsynchronized pseudonym changes on the degree of anonymity achieved. A bunch of efficient special purpose pseudonym constructions have been proposed, e.g. for RFID applications. Henrici et al. [29] propose a hash-based construction of pseudonyms, which allows for implementing changing pseudonyms. Gruteser et al. [26] also use hash-based constructions, called hash-chains, to construct short-lived pseudonyms. Comparable to our work, Juels et al. [32], use public key encryption and argue that this approach is efficient even for lightweight RFID applications, w.r.t. privacy protection. Notably, Heinemann [28] proposes to implement privacy-respecting opportunistic networks based on changing pseudonyms, whereby he proclaims that identifiers must be changeable within every communication layer. A different use of pseudonyms is presented by Delakouridis et al. [14], applying pseudonyms to the problem of storing and accessing location information in a privacy-preserving, decentralized manner. While our work shares the idea of using pseudonyms as reference points for data access, they propose to split the information to be protected according to Shamir secret sharing [47], and to distribute those shares on several servers, addressable via pseudonyms.

8.3 Privacy-Respecting Audit Log Analysis

The use of pseudonyms in audit logs was first suggested by Fischer-Hübner et al. [20]. Pseudonymous auditing is now a widely recognized approach to balance the conflicting security requirements of accountability and privacy, and often combined with intrusion detection systems, that automatically analyze pseudonymized logs [48, 21]. The analysis of pseudonymous logs inherently requires means for linking the pseudonyms to each other, and finally to users. Research in this direction is about finding efficient mechanisms and models of operations. Biskup et al. propose to use transaction pseudonyms [6] in audit logs and a secret sharing method for the re-identification [5] in case a threshold of detected inappropriate actions is exceeded in the analysis. These concepts are applied within traditional computer security domains like Unix audit logs. We extend the application perspective to real-world audit logs, that emerge from location tracking applications. Technically, we also propose a new approach building on efficient secure multiparty computation techniques that we apply on the pseudonym level. The inherent use of verifiable threshold cryptography and reencryption mixnets, techniques with a long tradition of research in the area of cryptographic protocols, also allows us to provide a second level of auditability, in order to achieve multilateral security.

9 Summary and Conclusions

In this chapter, we presented and discussed a novel approach for realizing multilaterally secure auditing functionalities. The approach is tailored to be employed in location tracking applications and Ubiquitous Computing scenarios. Especially, we motivated our work by collaborative mission-critical applications. In this context, we proposed to realize so called multilaterally secure mission logs, which actually extend the traditional concept of computer audit logs to real-world audit logs. Hereby, we take into account the trend that computers pervade more and more parts of our everyday life and work, which broadens the actual scope of logging and auditing possibilities and also *required functionalities*.

On a conceptual level, our work combines pseudonyms with secure multiparty computation, both classic examples in the realm of privacy-enhancing technologies. Nevertheless, making them work in real-world application scenarios is a challenging area of research. Our approach builds on the principle of *accountability by auditability*. We employ it on two conceptual layers, first regarding real-world actions of mobile users as well as regarding the use of auditing functionalities by audit officers. Additionally, we support the individual user by enabling her to repudiate false accusations by means of log attestation. We also take into account a broader legal perspective. The lawful disclosure functionality allows a single attorney authority to globally revoke privacy protection of individual users. Regulations in some application domains, e.g. that of emergency management, demand such a functionality. However, in the light of key escrow discussions [1], we stress that it is also possible to instantiate our scheme without that property. We believe that the presented approach and its flexibility to tailor it to concrete application needs is a step towards the realization of multilaterally secure thus multilaterally acceptable UbiComp systems. Addressing inherent tradeoffs already in the construction of technologies that pervade everyday life more and more is essential for their acceptance. The introduction of new technologies should always comply with legal and social backgrounds.

Our general approach to deal with this issue is to analyze concrete application scenarios. Mission-critical applications, along with their multilaterally demanding requirements, are an important example to learn how to design ICT support for a *seamless collaboration*.

Acknowledgements. This work was supported by CASED (www.cased.de). The authors are responsible for the content of this publication.

References

1. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B.: The Risks of Key Recovery, Key Escrow, and Trusted Third Party Encryption (1998), http://www.cdt.org/crypto/risks98/
2. Aitenbichler, E., Kangasharju, J., Mühlhäuser, M.: MundoCore: A Light-Weight Infrastructure for Pervasive Computing. Pervasive and Mobile Computing 3(4), 332–361 (2007)
3. Anderson, R.J.: Security Engineering: a Guide to Building Dependable Distributed Systems. John Wiley & Sons, Chichester (2008)
4. Beresford, A.R., Stajano, F.: Location Privacy in Pervasive Computing. IEEE Pervasive Computing 02(1), 46–55 (2003)
5. Biskup, J., Flegel, U.: Threshold-Based Identity Recovery for Privacy Enhanced Applications. In: ACM Conference on Computer and Communications Security, pp. 71–79. ACM, New York (2000)
6. Biskup, J., Flegel, U.: Transaction-Based Pseudonyms in Audit Data for Privacy Respecting Intrusion Detection. In: Debar, H., Mé, L., Wu, S.F. (eds.) RAID 2000. LNCS, vol. 1907, pp. 28–48. Springer, Heidelberg (2000)
7. Burmester, M., Desmedt, Y., Wright, R.N., Yasinsac, A.: Accountable Privacy. In: Security Protocols Workshop 2004, pp. 83–95. Springer, Heidelberg (2004)
8. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
9. Cas, J.: Privacy in Pervasive Computing Environments - A Contradiction in Terms? IEEE Technology and Society Magazine 24(1), 24–33 (2005)
10. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24(2), 84–88 (1981)
11. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)
12. Cramer, R., Damgard, I., Schoenmakers, B.: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)
13. Cramer, R., Gennaro, R., Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)
14. Delakouridis, C., Kazatzopoulos, L., Marias, G.F., Georgiadis, P.: Share The Secret: Enabling Location Privacy in Ubiquitous Environments. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 289–305. Springer, Heidelberg (2005)

15. Duckham, M., Kulik, L.: Location Privacy and Location-Aware Computing. In: Dynamic & Mobile GIS: Investigating Change in Space and Time, pp. 34–51. CRC Press, Boca Raton (2006)
16. ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)
17. Ferguson, N., Schneier, B.: Practical Cryptography. Wiley Publishing, Inc., Chichester (2003)
18. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
19. Fischer-Hübner, S.: Pseudonymity. In: Encyclopedia of Database Systems, p. 2207 (2009)
20. Fischer-Hübner, S., Brunnstein, K.: Combining Verified and Adaptive System Components Towards More Secure System Architectures. In: Workshop on Computer Architectures to Support Security and Persistance of Information. Springer, Heidelberg (1990)
21. Flegel, U.: Privacy-Respecting Intrusion Detection. Springer, Heidelberg (2007)
22. Freudiger, J., Manshaei, M.H., Hubaux, J.-P., Parkes, D.C.: On Non-Cooperative Location Privacy: a Game-Theoretic Analysis. In: ACM Conference on Computer and Communications Security, pp. 324–337. ACM, New York (2009)
23. Furukawa, J., Sako, K.: An Efficient Scheme for Proving a Shuffle. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 368–387. Springer, Heidelberg (2001)
24. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems. SIAM Journal on Computing 18(1), 186–208 (1989)
25. Greenstein, B., McCoy, D., Pang, J., Kohno, T., Seshan, S., Wetherall, D.: Improving Wireless Privacy with an Identifier-Free Link Layer Protocol. In: Conference on Mobile Systems, Applications, and Services (MobiSys 2008), pp. 40–53. ACM, New York (2008)
26. Gruteser, M., Grunwald, D.: Enhancing Location Privacy in Wireless LAN through Disposable Interface Identifiers: a Quantitative Analysis. Mob. Netw. Appl. 10(3), 315–325 (2005)
27. Hartmann, M., Austaller, G.: Context Models and Context-Awareness. In: Ubiquitous Computing Technology for Real Time Enterprises, pp. 235–256. IGI Global Publisher (2008)
28. Heinemann, A.: Collaboration in opportunistic networks. Ph.D. thesis, Technische Universität Darmstadt (2007)
29. Henrici, D., Müller, P.: Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers. In: Conference on Pervasive Computing and Communications Workshops (PERCOMW 2004). IEEE Computer Society, Los Alamitos (2004)
30. Hirt, M.: Multi-party computation: Efficient protocols, general adversaries, and voting. Ph.D. thesis, ETH Zurich (September 2001), ftp://ftp.inf.ethz.ch/pub/crypto/publications/Hirt01.pdf
31. Jakobsson, M., Juels, A.: Mix and Match: Secure Function Evaluation via Ciphertexts (Extended Abstract). In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 162–177. Springer, Heidelberg (2000)
32. Juels, A., Pappu, R.: Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)

33. Kesdogan, D., Federrath, H., Jerichow, A., Pfitzmann, A.: Location Management Strategies Increasing Privacy in mobile Communication. In: IFIP International Information Security Conference (SEC 1996), pp. 39–48. Chapman & Hall, Boca Raton (1996)
34. Koeune, F.: Pseudo-Random Number Generator. In: Encyclopedia of Cryptography and Security, pp. 485–487 (2005)
35. Lindell, Y., Pinkas, B.: Secure Multiparty Computation for Privacy-Preserving Data Mining. Journal of Privacy and Confidentiality 01(01), 59–98 (2009)
36. Martucci, L.A., Kohlweiss, M., Andersson, C., Panchenko, A.: Self-Certified Sybil-Free Pseudonyms. In: Conference on Wireless Network Security (WISEC 2008), pp. 154–159. ACM, New York (2008)
37. Mühlhäuser, M., Gurevych, I. (eds.): Ubiquitous Computing Technology for Real Time Enterprises - Handbook of Research. IGI Global Publisher (2008)
38. Mühlhäuser, M., Hartmann, M.: Interacting with Context. In: Rothermel, K., Fritsch, D., Blochinger, W., Dürr, F. (eds.) QuaCon 2009. LNCS, vol. 5786, pp. 1–14. Springer, Heidelberg (2009)
39. Park, C., Itoh, K., Kurosawa, K.: Efficient Anonymous Channel and All/Nothing Election Scheme. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 248–259. Springer, Heidelberg (1994)
40. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party (Extended Abstract). In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 522–526. Springer, Heidelberg (1991)
41. Pfitzmann, A.: Multilateral Security: Enabling Technologies and Their Evaluation. In: Müller, G. (ed.) ETRICS 2006. LNCS, vol. 3995, pp. 1–13. Springer, Heidelberg (2006)
42. Pfitzmann, A., Hansen, M.: A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. V0.32 (December 2009), http://dud.inf.tu-dresden.de/Anon_Terminology.shtml
43. Rannenberg, K.: Multilateral Security - a Concept and Examples for Balanced Security. In: Workshop on New Security Paradigms (NSPW 2000), pp. 151–162. ACM, New York (2000)
44. Satyanarayanan, M.: Privacy: The Achilles Heel of Pervasive Computing? IEEE Pervasive Computing 2(1), 2–3 (2003)
45. Schlott, S.: Privacy- und sicherheitsaspekte in ubiquitaeren umgebungen. Ph.D. thesis, Universität Ulm (2008)
46. Schneier, B., Kelsey, J.: Secure Audit Logs to Support Computer Forensics. ACM Trans. Inf. Syst. Secur. 2(2), 159–176 (1999)
47. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
48. Sobirey, M., Fischer-Hübner, S., Rannenberg, K.: Pseudonymous Audit for Privacy Enhanced Intrusion Detection. In: IFIP International Information Security Conference (SEC 1997), pp. 151–163. Chapman & Hall, Boca Raton (1997)
49. Stajano, F.: Security Issues in Ubiquitous Computing. In: Handbook of Ambient Intelligence and Smart Environments, pp. 281–314. Springer, Heidelberg (2010)
50. Tsiounis, Y., Yung, M.: On the Security of ElGamal based Encryption. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 117–134. Springer, Heidelberg (1998)
51. Waters, B.R., Balfanz, D., Durfee, G., Smetters, D.K.: Building an Encrypted and Searchable Audit Log. In: Network and Distributed System Security Symposium (NDSS 2004). The Internet Society (2004)

52. Weber, S.G.: Harnessing Pseudonyms with Implicit Attributes for Privacy-Respecting Mission Log Analysis. In: Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), pp. 119–126. IEEE Computer Society, Los Alamitos (2009)
53. Weber, S.G.: Securing First Response Coordination with Dynamic Attribute-Based Encryption. In: World Congress on Privacy, Security, Trust and the Management of e-Business (CONGRESS 2009), pp. 58–69. IEEE Computer Society, Los Alamitos (2009)
54. Weber, S.G., Heinemann, A., Mühlhäuser, M.: Towards an Architecture for Balancing Privacy and Traceability in Ubiquitous Computing Environments. In: Workshop on Privacy and Assurance (WPA 2008) at Conference on Availability, Reliability and Security (ARES 2008), pp. 958–964. IEEE Computer Society, Los Alamitos (2008)
55. Weber, S.G., Ries, S., Heinemann, A.: Inherent Tradeoffs in Ubiquitous Computing Services. In: INFORMATIK 2007. LNI, vol. P109, pp. 364–368. GI (September 2007)
56. Weiser, M.: The Computer for the 21st Century. Scientific American 265(3), 94–104 (1991)
57. Yao, A.C.: Protocols for Secure Computations (Extended Abstract). In: 23th Annual Symposium on Foundations of Computer Science (FOCS 1982), pp. 160–164. IEEE Computer Society Press, Los Alamitos (1982)

Intrusion Detection in Multi-Agent Systems

Bogdan Ghit, Florin Pop, and Valentin Cristea

Abstract. This chapter presents an adaptive intrusion detection system for distributed environments dedicated to developing agent-based applications. We propose a scalable, flexible and reactive agent based architecture and a lightweight genetic algorithm that recognizes the intruders in an adaptive and automatic way. Our approach is based on monitoring the level of physical resources usage and implies the detection of those agents that manifest an abusive behavior. We finally enhance Jade with our intrusion detection system and we analyze and illustrate the results obtained in different scenario cases.

1 Introduction

Intrusion detection is a major concern in any computer system and the capability of ensuring protection against different types of attacks or abuses proves to be a very complex and challenging problem.

Distributed environments, due to their dynamic evolution in time, with continuous changes in configurations, software implementations and usage patterns, require an adaptive view of systems security.

The permanent evolution of computer system's attacks and the various events that may occur in a distributed system, determined researchers to permanently develop and improve security approaches and models.

Intrusion detection systems are used for monitoring events occurring in computer systems and analyzing them in order to detect abusive behaviors of users that might try to compromise the system's performance, availability and integrity. Due to various types of attacks that a system may be exposed to, the problem of detecting intruders is very complex. It is impossible to create a system able to handle any kind of events in a straightforward manner.

We propose an innovative system for intrusion detection in distributed environments such as agent platforms that identifies and rejects abusive behaviors in order to maintain a high level of performance and availability.

Bogdan Ghit · Florin Pop · Valentin Cristea
Automatic Control and Computers Faculty,
University "Politehnica" of Bucharest

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 235–255.
springerlink.com

Our idea is to create an adaptive architecture based on flexible, reactive and proactive entities that protect the system against malicious users, by interacting and cooperating with each others. Though, we represented our intrusion detection system as a multi-agent system and we designed a genetic algorithm that identifies abnormal behaviors within the platform in terms of physical resources usage.

The genetic approach is meant to enhance the system with adaptive and self-learning capabilities, fact which ensures a defensive reaction in case of various abusive events and a broader view of what is occurring within the system.

The final goal of our work was to integrate the intrusion detection system that we designed into Jade, a FIPA-compliant agent framework and to test and analyze its performance with various scenario cases.

The structure of the chapter is as follows: first we present the current architectural models used in intrusion detection, we continue with a brief presentation of the available solutions for agent platforms and we emphasize the main features that convinced us to use Jade for developing our system. Afterwards, we highlight our contributions to this research field: the system architecture of an adaptive intrusion detection system, its main components and the way we have integrated them in Jade, the genetic algorithm designed specifically for detecting anomalies in agents behaviors, the results obtained in experiments and finally we draw the conclusions and the next targets of our research.

2 Related Work

As we shall see, our work is mainly related with two research fields: intrusion detection systems and agent platforms, technologies that we are going to present in the following section.

Detecting and rejecting entities within a computer system that executes potentially harmful activities represents an important concern in nowadays security research. An intrusion attempt is any threat to the confidentiality, availability, or integrity of the information on a given system. The intruder distinguishes itself from other entities through its actions that usually generate an abnormal behavior of the system. In other words, an intrusion detection system is an automated system that permanently monitors events occurring in a computer system or network in order to detect and reject intruders [11].

The process of detecting intruders is very complex and implies a set of functions such as: monitoring and analyzing system’s behavior, establishing a line between what represents good and bad behavior in the system, detecting abnormal activities and taking actions in order to prevent system's crash.

There are two main types of IDS: network-based IDS, which detect intruders by monitoring traffic through network devices and host-based IDS which monitor the activities related to a software environment associated with a specific host [4]. These types of systems dynamically inspect network packets and can detect the resources accessed by any program. We are going to focus on the latter, because our purpose is to create an intrusion detection system for an agent platform.

The most common approaches for IDS are based on misuse detection and anomaly detection [16]. The former refers to techniques that find methods and actions which may penetrate a system. These penetrations represent patterns or signatures (a static string or a sequence of actions) that the IDS should identify. Passive methods analyze network activities following only quantity criteria, by inspecting certain databases with different attack signatures [11], [18].

On the other hand, anomaly detection refers to techniques that define a line between normal and abnormal behavior of the platform (e.g. CPU usage, job execution time, system calls). Behaviors that deviate from the expected normal behavior are marked as intrusions. Considering the fact that an intruder usually takes a large number of exploits in order to compromise a system, an IDS should be able to identify suspicious activities after a certain threshold has been exceeded [11].

According to [8] and [17], an intrusion detection system should have the following characteristics: permanent monitoring, fault tolerance, minimal overhead on the system it is running, and adaptability. As the number of monitored systems increases very fast, the system should also present: scalability, dynamic reconfiguration (allows changing the configuration without restarting the whole system) and graceful degradation of service (failures of some components should not influence the whole system).

Another classification of intrusion detection systems refers to the distribution of their components: the centralized IDS, where the inspection of the collected data is realized in a limited number of nodes and the distributed IDS, where the data analysis is performed in a number of locations proportional to the number of monitored hosts. Even though a centralized approach has some limitations regarding scalability, it also has several advantages over the distributed IDS [8]:

- Requires a small number of active components
- Makes easier the recovery process after crashes
- Determines a small overhead on the systems, except the central component
- Detects easier the changes in the global behavior of the system.

Crosbie and Spafford [3] proposed an autonomous model based on agents that could be evolved using genetic programming in order to automatically adjust the IDS to user behavior. The architecture of such a model is represented by several components that together accomplish the role of intrusion detection with respect to the list of characteristics mentioned above:

- Agents, used for monitoring and reporting events to a transceiver
- Transceivers, responsible for controlling the local agents of the host and for executing commands received from monitors
- Monitors, with global control and data processing roles in the system.

Agents represent a key element of our research, as they endow the system with many features that contribute to a scalable, flexible and reactive intrusion detection system.

The Foundation for Intelligent Physical Agents [1], known as FIPA developed a collection of standards for agent technology that led to a set of specifications supporting inter-agent communication and key middleware services, a well-specified agent communication language (FIPA-ACL) and a set of interaction protocols ranging from single message exchange to much more complex transactions.

Agent technology is meant to offer an improved framework for developing smarter, more flexible and efficient applications in the field of distributed systems. Numerous approaches, toolkits, and platforms have been created in order to facilitate a generic, flexible, and scalable environment for building intelligent agents.

A representative example is Zeus MAS [10], [25] developed by British Telecom Labs, implemented in Java, compliant with FIPA standards, providing communication mechanisms based on KQML (Knowledge Query and Manipulation Language) or ACL (Agent Communication Language) and with strong security policies, including public-private key digital signature technology for authentication mechanism. The main principles that fulfill its creators philosophy require *an open and extensible design* that facilitates users to add to the library new agent level components and *an agent-level functionality* ensured by communication, coordination, task execution and monitoring mechanisms that allow developers to provide only the code necessary for solving domain-specific problems. The major drawback of this platform is the lack of mobility function for agents, which is an essential feature for our intrusion detection system, as we will see next.

Jack Intelligent Agent [12] includes a Java-based agent oriented development environment and several functionalities that recommend this technology as a platform for executing agents endowed with services for message marshalling and a name server necessary for the lightweight communication mechanism between agents. Even though this platform is also FIPA compliant and includes other functionalities such as the ability of designing hierarchical teams of agents, it does not provide any mobility mechanisms.

JADE [1], [22], [26] is an agent-based platform, full-compliant with FIPA specifications, which provides a set of basic middleware layer functionalities, easy to customize according to the programmers requests and used to improve and simplify the development of distributed systems. JADE is a fully distributed system inhabited with agents, each running as a separate thread, eventually on different remote machines, which are able to manage a transparent communication with each other. Certain features of JADE agents like having own threads of execution, taking autonomous decisions in a peer-to-peer system, being loosely coupled and identified by a globally unique name, recommend JADE as a representative platform for developing and analyzing agent-based distributed applications.

JADE architecture is based on the peer-to-peer model of communication and facilitates the distribution across a group of heterogeneous hosts of the complete set of services mentioned in FIPA specifications: the intelligence, the initiative, the information, the resources and the control. The evolution of the environment is dynamic due to agents' features which can join and leave the system at any moment. Moreover, two very important features for the intrusion detection system we aim to develop are supported by JADE: first, it provides agent mobility (even though it is able to migrate code only, without execution state, it is very helpful in ensuring agents autonomy); second, it also offers a replication service for the main container which is used as a fault tolerance mechanism for our centralized implementation of the IDS.

Lightweight Extensible Agent Platform [13], known as LEAP, is a development and run-time environment for Intelligent Agents and represents a major technical challenge, because it aims to become the first integrated agent development environment capable of generating agent applications in the ZEUS environment and executing them on run-time environments derived from JADE, implemented over a large family of devices (computers, PDA and mobile phones) and communication mechanisms (TCP/IP, WAP). In this way LEAP benefits from the advanced design-time features of Zeus and the lightweight and extensible properties of JADE.

Considering the fact that it has already been used in many development projects, it is fully-compliant with FIPA standards, it has very good security features and provides agent mobility and various communication protocols, Jade proves to be a proper environment for integrating and testing our IDS.

3 Intrusion Detection System for JADE Platform

JADE is a fully distributed system, with containers spread between multiple hosts, each of them being managed by users that may have access to different services offered by the platform. Any user may create its own agents and move them into one container running in a certain point of the network [22].

Furthermore, JADE may limit unauthorized access to the network with the security features through JADE-S add-on: the authentication process is required and also several permissions for agents belonging to a certain user may be established. In spite of this, there are still risks, even for the authorized and registered users, to compromise the platform from the point of view of the performance.

As in any distributed system, a major problem is to manage the system's physical resources and to ensure a convenient level of performance for the applications running on the platform. We should avoid the situations when one or more agents obtain the control of the processor or require a considerable amount of memory, because these might lead to performance penalties and other critical applications may suffer. As long time as we don't have any control over the agents created or migrated by a certain user, we cannot predict what are the real intentions for joining the platform, but we may observe the changes that occur in processor's activity, memory usage or other requested services. In this manner we may establish the line between what means a proper behavior or state of the platform and what means a harmful and bad behavior of the platform.

The permanent growing and expansion of the computer networks and their complex and dynamic behaviors make impossible the detection of any intruder with a single general rule that reflects all the resources and services parameters involved in an agent platform. We need an adaptive and self-learning model able to analyze platform's state in order to detect and reject the intruders.

According to all these presented aspects, the IDS systems can be improved by using genetic algorithms. The IDS can be viewed as a rule-based system and genetic algorithms can be viewed as a tool to help generate knowledge. Saniee et al [27] proposed a parallel genetic local search algorithm (PAGELS) to generate fuzzy rules capable of detecting intrusive behaviors in computer networks. In the presented algorithm the global population is divided into some subpopulations, each assigned to a distinct processor. Each subpopulation consists of the same class fuzzy rules. These rules evolve independently in the proposed parallel manner. Another approach is based on information processing based on data extraction [28]. In that way the high level of detection rates of attacks is kept while speeding up the processing of the data.

The approach offered by the genetic algorithm is a convenient one and fits perfectly to the dynamic aspects which occur during a platform's cycle of life. Our purpose is to create an IDS based on lightweight, diversified, autonomous agents that work separately and cooperate in order to identify and eventually remove those elements in the system that determine an abnormal behavior, in the same manner like the human immune system acts. These agents have a strong intelligent component which analyzes the reaction of the platform to different system activities and tries to detect and learn the malicious actions by monitoring and communicating with each others.

3.1 Intelligent Intrusion Detection System

Early models proposed for IDS have been viewed as a problem of pattern classification, but the system's intrinsic characteristics make it very difficult to detect intrusions directly, using classical pattern recognition methods. Both, the distributed platform and the natural human immune system cannot recognize patterns with linear methods, but the interesting part is that the second one successfully protects our body against a vast variety of foreign pathogens [20], [21].

The idea of developing an intrusion detection system based on self-adapting and self-learning mechanisms borrowed from the human immune system was proposed by Jeffrey Kephart in [23], [24].

The architecture of an Intelligent Intrusion Detection System combines agents' reactivity, mobility and autonomy with artificial intelligence techniques which are used to expand the solutions space and offer more accurate results. A bio-inspired intrusion detection system should be able to employ multi-level defense mechanisms for protecting the system against a wide variety of attacks [20].

A genetic algorithm monitors network connections and translates its related behaviors to represent a rule in order to judge if a real-time connection may be considered or not an intrusion. These rules can be modeled as chromosomes inside the population which evolves until the evaluation condition is met. The generated rule set can be used as knowledge inside the IDS for judging whether the network connection and related behaviors are potential intruders [15], [17], [19].

3.2 IDS Integration in Jade

We propose a distributed monitoring process, with centralized data analysis, based on autonomous agents living inside Jade that use a genetic algorithm for detecting intruders that abuse system's physical resources [14].

For a more specific presentation of the IDS's architecture, we are going to illustrate several details of JADE. JADE platform is composed of agent containers that can be distributed over the network. Agents live in containers, Java processes that provide the Jade run-time and all the services needed for hosting and executing agents. There is a special container, called the main container, which represents the bootstrap point of a platform: it is the first container to be launched and all other containers must join to a main container by registering with it. The single point of failure is easily handled by the Main Replication Service, fact which ensures that the platform remains fully operational even in the event of a main-container failure. When the main-container is launched, two special agents are automatically instantiated and started by JADE:

1. The *Agent Management System* (AMS) is the agent that supervises the entire platform. It is the contact point for all agents that need to interact in order to access the white pages of the platform as well as to manage their life cycle. Every agent is required to register with the AMS (automatically carried out by JADE at agent start-up) in order to obtain a valid AID.

2. The *Directory Facilitator* (DF) is the agent that implements the yellow pages service, used by any agent wishing to register its services or search for other available services. The DF also accepts subscriptions from agents that wish to be notified whenever a service registration or modification is made that match some specified criteria. Multiple DFs can be started concurrently in order to distribute the yellow pages service across several domains. These entities can be federated, if required, by establishing cross-registrations with one another which allow the propagation of agent requests across the entire federation.

Each module illustrated in Fig. 1 is represented by one or more Jade agents that accomplish the requested goals for an intrusion detection system. As we can observe, the architecture is a centralized one, the main process of detecting the intruders being represented by a Manager Agent that communicates with its explorers in order to monitor and execute the actions for keeping an optimal level of performance in the platform.

The *Monitoring Layer* is represented by those agents migrated by the Manager Agent in each living container in order to monitor and send information about the physical resources usage in the container they have been launched. These notifications are sent continuously, after a predefined timeout expires. Each *Monitor Agent*, before starting the execution of its tasks, has to create an *Action Agent* ready to execute commands received from the manager (for instance, terminate an agent or a set of agents).

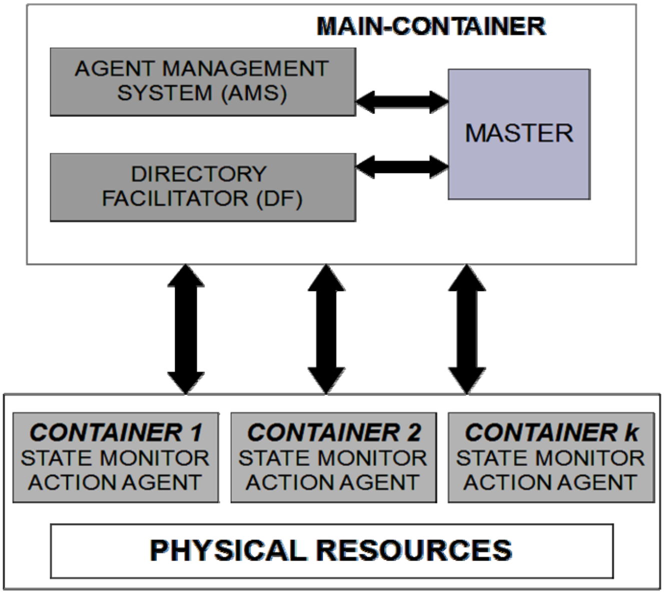


Fig. 1 IDS Integration in JADE Platform

The *Services Layer* is represented by the AMS and DF, which are two important entities in JADE platform. All agents implemented in the modules of the intrusion detection system use the services offered by those two agents, because they need to publish their services, to discover other agents' services or to be notified when different platform's events occur.

The *Detection Layer* is implemented as one of the *Manager Agent*'s functionalities and it is provided by the genetic algorithm that offers an immune approach for our intrusion detection system. It helps us to determine the set of actions that must be applied in order to increase system's performance, so in the final stage of detection it has to execute the commands that complete those actions. For this reason, it sends a kill-message to the *Action Agent* living in the same container with the agent selected from the set of actions.

The *Manager Agent* communicates with the *Services Layer*, because it has to register its own service and also to search the agents that will execute the set of actions resulted from the genetic analysis and also receives event notifications from the AMS. It is obviously connected to the *Detection Layer*, because it is the one that evaluates the global use of physical resources in the platform and takes the decision of launching the genetic algorithm in order to detect the intruders when a certain threshold is reached.

A very important issue that has to be mentioned and discussed is the one about the fault tolerance. We have seen from the system's architecture that the proposed solution is a centralized one, with a single point of failure which is the *Manager Agent*. This agent is critical for the intrusion detection process and without it, we don't have access to the platform's state and we cannot control and analyze agents' behavior, which means that the intruders are free to access and compromise the system's performance. Fortunately, JADE offers two services in order to handle the fault tolerance problem: the first one provides the main container replication which allows the launching of multiple main-containers as replica backups and the second one guarantees Directory Facilitator's persistence by storing its catalogue of yellow pages into a relational database.

The intrusion detection system has a complex architecture based on three connected modules that communicate and cooperate in order to achieve the goal of creating an immune defensive mechanism for the platform.

4 Genetic Algorithm for Intrusion Detection

Genetic algorithms [15] are a family of computational models based on concepts of natural evolutionary systems. These algorithms encode a potential solution to a specific problem as a chromosome and perform recombination operators on this data structure to evolve new solutions. The reproduction process is performed in such a way that the chromosomes representing better solutions are given more chances to reproduce than those representing worse solutions. The goodness, also called fitness is defined according to the desired property of the final solution.

A genetic algorithm starts with a randomly generated population or in some situations with a predefined initial population of chromosomes encoding a potential solution of the problem, evolves through selection, recombination, mutation or other genetic operators and returns the best chromosome when the optimization criteria is met or the maximum number of generations is reached. The mechanism of a genetic algorithm is shown in Fig. 2.

Very difficult real-world problems can be optimized with such techniques that simulate the natural evolutionary process, and most of the time, their results outperform conventional methods [29].

Security is one of the computer fields where genetic algorithms gain more and more interest, mainly due to their ability of exploring the search space in an adaptive and self-learning manner.

An intrusion detection system is often associated with a rule based system [6], which stores certain rules following the structure below, that are used to differentiate normal behavior from anomalous behavior of the system.

if {conditions} then {actions}

The idea is to store a set of rules into a database and monitor the state of the system in order to detect the moment when certain conditions are met. At this point the list of actions indicated by the corresponding stored rule is applied and the system's integrity is ensured. The purpose of the genetic algorithm is obvious, as these rules cannot be predefined or statically stored, which means that we need a dynamic mechanism for generating them.

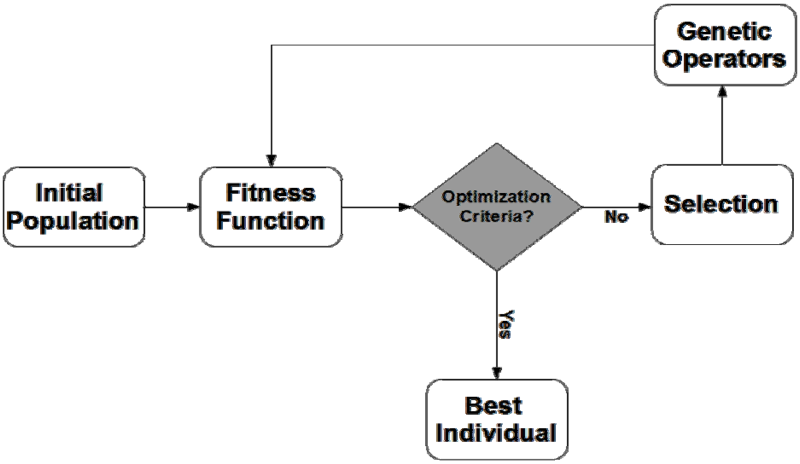


Fig. 2 Genetic Algorithm Steps

Current genetic approaches for intrusion detection systems are based on simple rules for network traffic that are used to detect suspicious connections [6]. In this situation, a genetic algorithm will evolve the rules stored in the database at a certain moment in order to obtain new and possible more accurate rules for filtering the traffic based on the existing knowledge on current connections.

Conditions may contain restrictions regarding the source IP address, destination IP address, port numbers, connection time or other TCP/IP properties, while the actions may imply stopping the connection or blocking a certain user that satisfies these conditions [3].

Algorithm's effectiveness is influenced by many parameters (evaluation function, mutation and crossover rates, number of generations, initial population) that should be considered and adjusted according to the application environment in order to obtain a fast convergence through the solutions space.

The problem of obtaining system's immunity through intrusion detection in an agent platform is quite different from the previous approach. As long as we do not have any control over the agents created or migrated inside the platform by different users, we are not able to predict the actions that are going to be executed, but we can observe the changes that occur in processor's activity, memory usage or other requested service. These are important elements that help us to establish the line that separates the proper behavior of the system from the anomalous behavior of the system.

The permanent expansion of distributed systems and their complex and dynamic behavior make impossible the detection of any possible intruder with a single general rule that reflects all the parameters that describe the system's state. We need an adaptive and self-learning model capable of analyzing various behaviors of the platform in order to detect and reject possible intruders. The genetic approach is a convenient one and fits perfectly with the dynamic aspects that occur during an agent platform's cycle of life.

4.1 Chromosome Layout

When we define the chromosome layout we have to consider the fact that these data structures encode potential solutions for the problem of intrusion detection. A chromosome reflects the current state of the platform, so it will contain certain parameters that indicate the level of physical resources usage.

Following the principle of a rule-based system, we consider a database of rules composed by a set of conditions and another set of actions that must be applied in order to reestablish a previous state of the platform when its behavior was normal.

The solution of the genetic algorithm has to identify those agents inside the platform that make abuse of the system's physical resources. There are two elements that contribute to the chromosome structure, the conditions and the actions. Thus, we use the concept of supergene, which represents a group of neighboring genes on a chromosome inherited together due to their genetic linkage.

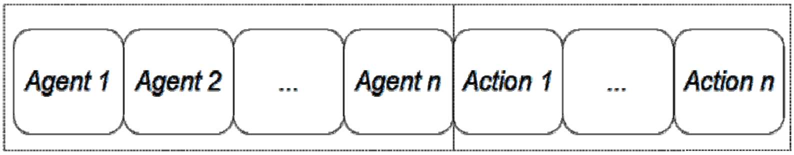


Fig. 3 Chromosome encoding

As we can notice from the structure illustrated in Fig. 3, the first supergene contains a set of genes encoding identifiers of agents that are currently executing several actions within the platform with potential negative influence over the system's performance.

The second supergene encodes another set of genes, representing codes associated with certain actions that must be applied: stop, kill or pause the agents that appear in the first supergene, ban or refuse a certain user interaction across the platform. All genes are integer numbers (agent identifiers or action codes) and the sizes of the chromosomes may vary.

4.2 Initial Population

Although generally it is not recommended to start with a random population when you design a genetic algorithm, for this problem there is no mistake to consider a stochastic generation.

The reason is simple: we do not know how to choose an initial population, as there is no available information which might help us generate some possible sets of actions to apply, because the system is invariant and has no guidance rules for its activity.

The potential solutions may have different number of agents inside the conditions supergene, so the population chromosomes may have different sizes. Both the initial number of individuals and their sizes are generated randomly or based on the total number of agents living within the platform.

4.3 Fitness Function

The evaluation function or the fitness function is the instrument that helps the genetic algorithm to search and find the best solution and reflects the *goodness* of a chromosome according to certain optimization criteria.

The fitness function for intrusion detection problem should express the increase of performance obtained by applying the actions encapsulated in the evaluated chromosome. Each component agent is analyzed in terms of its current physical resources usage (CPU usage and memory usage), and depending on the negative influence it has over the platform's performance, the corresponding action included in the second supergene of the chromosome may lead to the termination of the agent or to the suspension of the agent for a certain timeout.

The general expression of the fitness function represents a sum of two other functions which reflect the performance achieved by applying the set of actions over the existing agents inside the chromosome and on the other hand, the confidence in the evaluated individual.

$$fitness = W_1 \times performance + W_2 \times confidence \tag{1}$$

The first element represents the performance gain that is obtained by applying the set of actions over the agents encoded into the chromosome. It is calculated based on the CPU usage and memory usage of each agent from the evaluated solution as follows:

$$performance = C_1 \times \sum_{i=1}^{N} f(cpu_i) + C_2 \times \sum_{i=1}^{N} f(mem_i) \tag{2}$$

There are two types of actions that can be applied on agents: one is to terminate or kill the agent which basically removes the job executed by that agent and the other one is to stop or block the agent for a certain period of time. This period of time naturally has an influence on the performance gain obtained by stopping that agent, because once the timeout expires, the agent will continue its execution and will consume those resources that have been saved before. Thus, the benefits of a certain action are computed according to the next expression of function *f*:

$$f(x_i) = \begin{cases} x_i, action(i) = kill \\ \dfrac{T}{T_{\max}} \times x_i, action(i) = stop(T) \end{cases} \tag{3}$$

If the action associated with an agent is *killing* then the value of the resource remains unchanged, otherwise, in the case of *stopping* the agent for a period of time T, the value of the resource is diminished with a factor $\frac{T}{T_{\max}}$, where $T_{\max}$ is the maximum value of the timeout.

The second element of the fitness function controls the number of actions encoded in a certain chromosome: the higher the number of agents representing potential intruders, the lower the confidence in that particular rule.

This means that there has to be a tradeoff between the number of agents encapsulated into a chromosome (limited to a predefined value $N_{\max}$) and the performance gain that we expect to obtain by applying the corresponding actions. In this manner, we avoid solutions that include a large number of agents even though they do not execute any suspicious activity.

$$confidence = 1 - \frac{N}{N_{\max}} \tag{4}$$

Other parameters, such as W_1, W_2, C_1, and C2 should be initialized according to the importance of the weighted parameter and their values influence the convergence of the algorithm to the optimal solution.

4.4 Genetic Operators

The final step in designing a genetic algorithm is to define the genetic operators which have the major role of simulating the natural recombination process in order to evolve new generations of better individuals.

The selection process that chooses the individuals for reproduction is performed using a weighted roulette-wheel which assigns to each chromosome a section proportional with its fitness value. The higher the fitness, the higher the chance to evolve, fact which means better solutions tend to survive through the selection process.

We developed three genetic operators, one for mutation, one for deletion, and another one for crossover with the requirement of maintaining the chromosome’s integrity after their application. A chromosome is considered to be valid when the following conditions are met:

- it must have distinct genes, as the set of actions that can be applied for an agent are mutually exclusive
- the replacing gene must contain an identifier of a current registered agent

The application of a mutation operator implies two steps: it replaces a gene that encodes an agent identifier and also replaces the gene that encodes the action associated with the replaced agent, as we can observe in Fig. 4. The new action is defined according to the amount of physical resources consumed by the new agent; a threshold decides whether the action is *killing* or *stopping*: for instance if the agent CPU and memory usage reach 40%, then the action is termination, otherwise the action is stopping the agent for a period of time proportional with its level of resources usage.

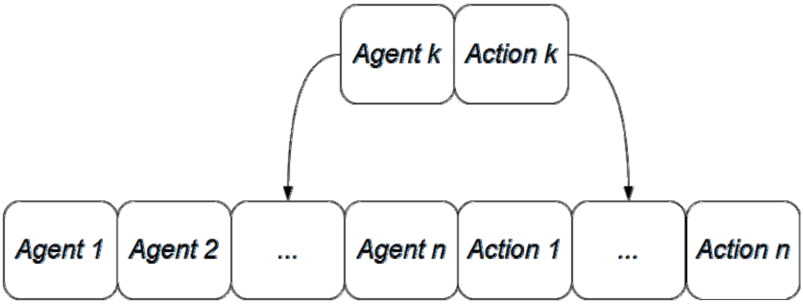


Fig. 4 Action of mutation operator

Considering the fact that the chromosome may have different sizes, we also need an operator able to remove certain genes of the chromosome during the evolutionary process.

This operator randomly selects a gene encoding a certain agent identifier and deletes it together with the gene encoding the associated action. In this manner, the diversity of the population in terms of chromosomes sizes is guaranteed. The deletion operator also contributes to the elimination of bad genes, fact that blocks the inheritance of weak features from the ancestors.

The third operator is the crossover that simulates the mating process from nature: two individuals, selected as parents create offspring by exchanging chromosome patterns between them (see Fig. 5).

The various sizes of the chromosomes determine the need of a *cut and splice* crossover, which selects two random genes from the chromosomes chosen for recombination and exchanges the resulted segments in order to generate new individuals. In the context of our chromosome representation, this operator implies a reorganization of the genes so that the original structure composed of two supergenes remains unchanged.

The crossover points are selected from the first supergene and after the exchange is realized between these parts of the chromosomes, the second supergenes are attached to both children.

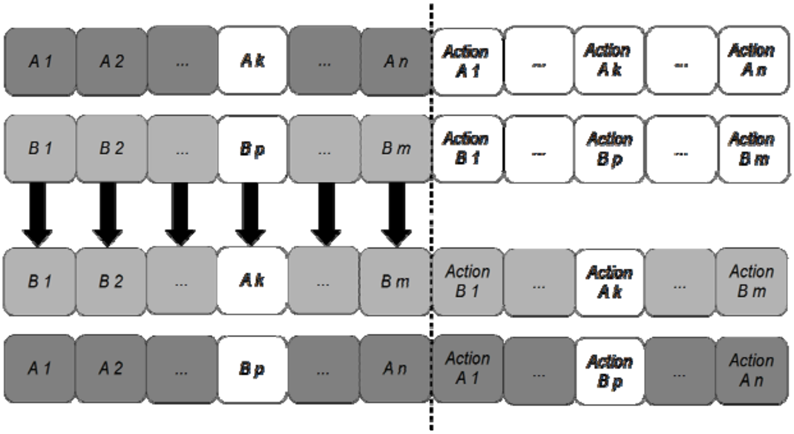


Fig. 5 Crossover operator

4.5 Discussion

As our purpose was to develop a lightweight solution, fundamental different from other approaches that use a historical knowledge, we obviously had to handle a trade-off: the performance obtained by using a small amount of information for intrusion detection in balance with the situation when the information gathered is not enough as agents may be resource-intensive, but still valid.

Even though the lack of history seems to be a disadvantage, there are many autonomous applications based on multi-agent systems without resource-intensive behaviors, but that are supposed to provide critical decisions in a small amount of time. The most relevant example is that of a multi-agent medical system for child care, presented in [30]. If this system accidentally hosts a CPU intensive agent, this might easily lead to a performance penalty that slows down the expected medical decisions.

Our intrusion detection system suits very well in this situation, because it filters agents based on their level of resources usage. Thus, there is no need to store information about system's behavior in time, as the intruders can be immediately detected based on the current state of the system. Furthermore, the algorithm allows us to establish a threshold that defines the accepted limit of resources usage. Depending on the applications the multi-agent system deals with, we can set this value in order to obtain more accurate results according to our expectations.

5 Test Scenarios and Experimental Results

There are a few steps that should be done in order to install the intrusion detection system. First, we have to start the JADE platform which is running the

Main-Container, with the default agents RMA (Remote Management Agent, a graphical tool used to interact with the platform), AMS and DF. Then we have to create the IDS Manager Agent which receives notifications about each container's state from the explorers (monitoring agents). In every container of the platform we run a monitoring agent which first creates an agent responsible for terminating the agents ordered by the manager. Afterwards, it starts the monitoring process of the container during which it sends reports to the manager.

The scenarios used for testing are made with one and three containers attached to the main-container of the platform. In each case we started several normal agents and a few acting like intruders.

The current level of resources usage is defined as the average between the CPU and the memory usage. These are two of the most important parameters that describe the resources usage of an application in a distributed system.

Moreover, our aim was to develop an intrusion detection system able to identify potential intruders without performing intensive processing of any historical data. So, instead of analyzing a database of rules or signatures that characterize the system's behavior at different moments in the past, we use only the current state of the system in order to identify the intruders.

The considered value for the threshold is 0.40. Regarding the constants that appear in the fitness function, we considered that both CPU and memory components have the same importance so $W_1 = 0.50$, while the influence of the confidence term should be lower, so the constant W_2 is 0.30.

The first case considered is the one with a single container, in which we start several normal agents (dummy agents, with no intensive computation or memory requirements) and after that three harmful agents, two of them making abuse of the processor time and the last one consuming a considerable amount of memory.

We expect that the genetic algorithm finds the three agents as soon as the threshold is reached and initiates the termination process for each of them. Any other agent in the platform must be avoided, as it doesn't have a suspicious activity.

The entries from the tables have the following meaning: *t* is the time measured in seconds since the monitoring phase began, while *Processor*, *Memory* and *Average* reflect the usage of physical resources in percentages; the set A_i represents normal agents, C_i are associated with CPU intensive agents and M_i are agents with high memory usage.

We consider a sampling time of 10 seconds that represents the frequency of the monitoring process. This means that at every 10 seconds, the monitoring agents gather and report back to the manager the state of the containers they belong to. The manager will run the intrusion detection algorithm in order to detect if any intruder occurred since the last sampling moment.

The platform's evolution illustrated in Fig. 6 and Table 1 is the following:

- The first step is for launching the platform and starting the intrusion detection system. No harmful activity detected.
- In the next two steps, we start two dummy agents with uninteresting activity.

- At step four we start an agent that requires more and more of the CPU time C_1.
- Starting with record five, we notice the increase of the CPU usage due to C_1. We start another similar agent C_2 and an agent with high memory usage M_1.
- At step six, both the CPU and memory usage reached considerable values that allow the manager to run the genetic algorithm which detects the three abusive agents. The set of actions is given in the same order as the fittest chromosome contains them.
- After terminating the agents, the level of resources usage is restored to the initial value.

Table 1 Single Container Case

No	Time	Processor	Memory	Average	Comments
1	4	0.00	0.03	0.01	Start JADE with IDS
2	14	0.00	0.04	0.02	Start A_1, A_2, A_3
3	44	0.00	0.04	0.02	Start A_4
4	54	0.00	0.22	0.11	Start C_1
5	64	0.38	0.22	0.30	Start C_2, M_1
6	74	0.93	0.67	0.80	Kill(M_1,C_2,C_1)
7	84	0.00	0.01	0.05	OK

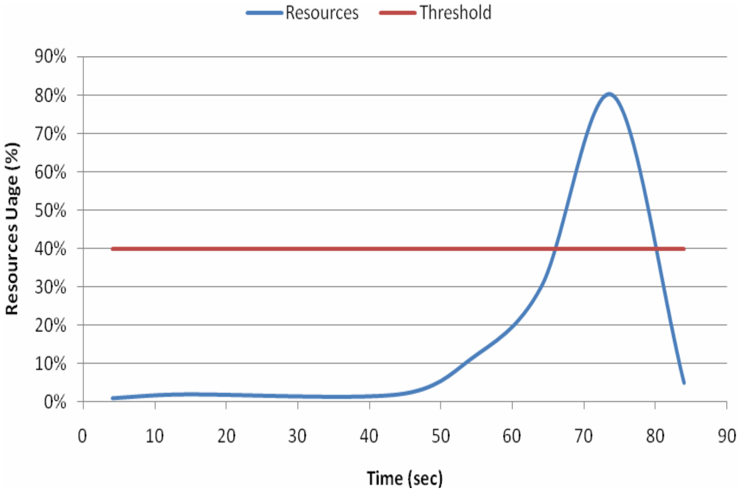


Fig. 6 Single Container Case

The second case considers a scenario with two containers having four normal agents and three intruders (see Fig. 7 and Table 2):

- In the first steps, it is launched the platform and the initial configurations are completed. Several agents with normal activity are migrated within the three containers.
- Then, between moments 53 and 72, in containers 1 and 3 appear two agents with high memory usage and in container 2 another agent that abuses the CPU.
- When the threshold of 0.40 is reached, those three abusive agents are detected and rejected by the IDS and the initial behavior is reestablished.
- The simulation continues, recording the appearance of other two agents both with CPU intensive activity, fact which leads to their termination.

Table 2 Three Containers Case

No	Time	Processor	Memory	Average	Comments
1	13	0.00	0.02	0.01	Start JADE and IDS
2	22	0.00	0.02	0.01	Start A_1-A_6 in Container-1,2,3
3	53	0.00	0.04	0.02	Start M_1 in Container-2
4	63	0.00	0.18	0.09	Start C_1 in Container-1
5	72	0.10	0.18	0.14	Start M_2 in Container-2
6	81	0.49	0.18	0.33	-
7	82	0.49	0.33	0.41	Kill(C_1, M_1, M_2)
8	92	0.00	0.01	0.00	OK
9	93	0.00	0.01	0.00	Start C_1 in Container-1
10	102	0.14	0.02	0.08	Start C_2 in Container-3
11	113	0.97	0.02	0.49	Kill(C_2, C_1)
12	123	0.00	0.01	0.00	OK

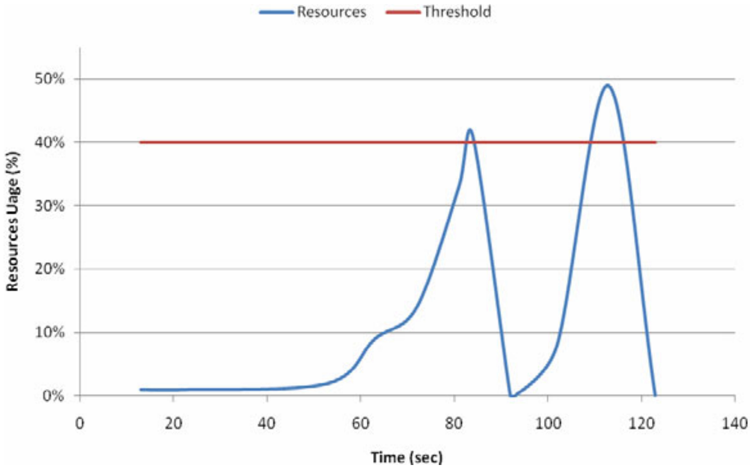


Fig. 7 Three Containers Case

The malicious agents have a progressive resources usage: at the beginning they have a normal behavior, but the longer they stay in the platform, the more resources they require, fact which usually happens with real intruders.

This scenario is illustrated in Fig. 7 where the appearance of the malicious agents leads to significantly longer time for detection than the earlier case. The detection time is also influenced by the sampling time (10 seconds in our experiments).

6 Conclusions and Future Work

We have developed an intrusion detection system for a distributed environment represented by an agent framework based on artificial intelligence techniques inspired from the human immune system. Our main contributions are the reactive agent-based architecture and the genetic algorithm that endow Jade with an adaptive intrusion detection system that monitors and rejects agents with a high level of physical resources usage.

The monitoring layer is an important component because it provides information and input data for the genetic algorithm. First, its results are used for estimating the general level of resources usage in the platform and then, according to the thresholds established, the Manager Agent may take the decision of running the genetic algorithm in order to find out the suspect agents.

Our solution is sustained by several test scenarios that illustrate the platform's state at different moments at its evolution in time and highlights the IDS reaction in case potential intruders occur.

As future work, we want to improve the intelligence of the Manager Agent in order to create a complete immune system, able to defend itself more accurately from any kind of malicious activity.

References

1. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE: A software framework for developing multi-agent applications (2008)
2. Balasubramaniyan, J.S., Garcia-Fernandez, J.O., Isacoff, D., Spafford, E., Zamboni, D.: An architecture for intrusion detection using autonomous agents. In: Proceedings of the 14th IEEE Computer Security Applications Conference (1998)
3. Crosbie, M., Spafford, G.: Applying genetic programming to intrusion detection. In: Working Notes for the AAAI Symposium on Genetic Programming, pp. 1–8 (1995)
4. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion detection using sequences of system calls. Journal of Computer Security 6(3), 151–180 (1998)
5. Kim, J., Bentley, P.J.: Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. In: Proceedings of the 2002 Congress on Evolutionary Computation, pp. 1015–1020. IEEE Press, Piscataway (2002)
6. Li, W.: Using genetic algorithm for network intrusion detection. In: Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference, Kansas City, Kansas, pp. 24–27. Citeseer (2004)

7. Nguyen, G.T., Dang, T.T., Hluchy, L., Balogh, Z., Laclavik, M., Budinska, I.: Agent platform evaluation and comparison. II-SAS, Pellucid EU 5FP IST-2001-34519 RTD, Technical report (June 2002)
8. Spafford, E.H., Zamboni, D.: Intrusion detection using autonomous agents. Computer Networks 34(4), 547–570 (2000)
9. Xu, H., Shatz, S.M.: Adk: An agent development kit based on a formal design model for multi-agent systems. Automated Software Engineering 10(4), 337–365 (2003)
10. Nwana, H.S., Ndumu, D.T., Lee, L.C., Collis, J.C.: ZEUS: a toolkit and approach for building distributed multi-agent systems. In: Proceedings of the Third Annual Conference on Autonomous Agents, pp. 360–361. ACM, New York (1999)
11. Bace, R., Mell, P.: Intrusion detection systems. National Institute of Standards and Technology (NIST), Special Publication 800-31 (2001)
12. Winikoff, M.: JACKTM intelligent agents: An industrial strength platform. In: Bordini et al. [5], pp. 175–193
13. Bergenti, F., Poggi, A.: Leap: A fipa platform for handheld and mobile devices. LNCS, pp. 436–446 (2002)
14. Ghit, B., Voicu, O., Pop, F., Cristea, V.: Distributed Agent Platform with Intrusion Detection Capabilities. In: International Conference on Intelligent Networking and Collaborative Systems, INCOS 2009, pp. 81–86 (2009)
15. Gen, M., Cheng, R.: Genetic algorithms and engineering design. Wiley-Interscience, Hoboken (1997)
16. Mukherjee, B., Heberlein, T.L., Levitt, K.N.: Network intrusion detection. IEEE Network 8(3), 26–41 (1994)
17. Crosbie, M., Spafford, G.: Active defense of a computer system using autonomous agents. Technical Report 95-008, COAST Group, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398 (February 1995)
18. Mosqueira-Rey, E., Alonso-Betanzos, A., Rio, B.B., Pieiro, J.L.: A Misuse Detection Agent for Intrusion Detection in a Multiagent Architecture. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 466–475. Springer, Heidelberg (2007)
19. Al-Ibaisi, T., Abu-Dalhoum, A.E., Al-Rawi, M., Alfonseca, M., Ortega, A.: Network intrusion detection using genetic algorithm to find best DNA signature. WTOS 7(7), 589–599 (2008)
20. Watkins, A.: An immunological approach to intrusion detection. In: Proceedings of 12th Annual Canadian Information Technology Security Symposium. Citeseer (2000)
21. Dasgupta, D., Gonzalez, F.: An immunity-based technique to characterize intrusions in computer networks. IEEE Transactions on Evolutionary Computation 6(3), 281–291 (2002)
22. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley Series in Agent Technology. John Wiley & Sons, Chichester (2007)
23. Kephart, J.O.: A biologically inspired immune system for computers. In: Proceedings of 4th International Workshop on Synthesis and Simulation of Living Systems, Cambridge, Massachusetts, USA, pp. 130–139 (1994)
24. Kephart, J.O., White, S.R.: Directed-graph epidemiological models of computer viruses. In: Computation: the micro and the macro view, p. 71 (1992)
25. Collis, J.C., Ndumu, D.T., Nwana, H.S., Lee, L.C.: The ZEUS Agent Building Toolkit. BT Technology Journal 16(3), 60–68 (1998)

26. JADE A FIPA-compliant agent framework. Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa (2008)
27. Saniee Abadeh, M., Habibi, J., Barzegar, Z., Sergi, M.: A parallel genetic local search algorithm for intrusion detection in computer networks. Eng. Appl. Artif. Intell. 20(8), 1058–1069 (2007)
28. Banković, Z., Stepanović, D., Bojanić, S., Nieto-Taladriz, O.: Improving network security using genetic algorithm approach. Comput. Electr. Eng. 33(5-6), 438–451 (2007)
29. Nicosia, G., Rinaudo, S., Sciacca, E.: An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization, Knowledge-Based Systems. In: The 27th SGAI International Conference on Artificial Intelligence, AI 2007, vol. 21(3), pp. 175–183 (April 2008) ISSN 0950-7051
30. Mago, V.K., Devi, M.S.: A Multi-agent Medical System for Indian Rural Infant and Child Care. In: Int. Joint Conference on AI, pp. 1396–1401 (2007)

A Round-Based Cover Traffic Algorithm for Anonymity Systems

Marta Rybczyńska

Abstract. Anonymity is becoming more of an issue with the growing importance of networking. Examples include protecting privacy of Internet users or electronic voting. Several network anonymity systems have been deployed, the most popular of them is probably Tor. However, those systems do not protect against attackers who observe or modify the traffic to match sources with destinations. The protection method against such attacks by adding additional traffic is not usually implemented because of the high cost. In this work we propose a new cover traffic generation algorithm for flow-based anonymity systems and compare it with other algorithms from the literature. Our algorithm is based on four ideas: fixed time rounds, flow classification with different protection methods for different classes, protection depending on the potential cost and finally, use of history. We evaluate our algorithm both in theory and in practice. Our analysis show that that our solution provides sufficient protection while reducing overhead traffic compared to the algorithms known from the literature.

1 Introduction

There are many definitions of anonymity. A short definition can be found in the anonymity terminology collected by Pfitzmann and Köhntopp [18], where anonymity is defined as the state of being not identifiable within a set of possible subjects (also called the anonymity set). Anonymity system that provides better anonymity to its' users is typically built as an overlay network over standard Internet architecture.

Marta Rybczyńska
Institute of Telecommunications,
Warsaw University of Technology,
Warsaw, Poland
e-mail: marta@rybczynska.net

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 257–280.
springerlink.com

Anonymity systems use various techniques to protect users' privacy, such as cryptographic transformation, introducing artificial delays, and reordering or adding additional (or cover) messages. Impressive effort has been put into improving the designs, but deployed systems are still vulnerable to certain attacks: protective mechanisms are either incomplete or require too many resources to be practical. A recent overview of existing designs and deployed systems may be found in a survey by Danezis and Diaz [6].

So called 'timing attacks' (or 'timing analysis') are probably the most important class of such attacks. The attacker observes messages in the network and searches for timing correlations to determine communication paths. Such attacks can be either active or passive.

In this work, we propose a new algorithm for cover traffic generation that provides decent protection against timing analysis while limiting the protection cost compared to existing solutions. Our algorithm also offers possibilities of even better protection. We compare our proposal with other algorithms from the literature. Our results are based on a theoretical analysis and simulations of a set of nodes with different traffic distributions, including a set of effective attacks proposed by other authors.

We begin by providing background on cover traffic algorithms in anonymity systems, known protections, attacks and the methods of calculating anonymity (Section 2). We model our anonymity system and attackers in Section 3. Section 4 presents our algorithm. We start from basic concepts, then move to the core algorithm and possible extensions. In Sections 5 and 6, we evaluate our algorithm. First we perform a theoretical evaluation using the degree of anonymity metric. Later we present and discuss simulation results in terms of protection and cost. We conclude in Section 7.

2 Previous Work

Cover traffic is additional traffic added into an anonymity system to make it harder for an attacker to trace flows in the system by making them more similar to each other. There is significant literature on the subject, but the deployed and popular anonymity systems like Tor do not use this protection method because of the high cost [9]. We review cover traffic in the first part of this section.

The second part is devoted to the methods of evaluating the protection level given by an anonymity system. We provide details on the most popular metric, the degree of anonymity.

2.1 Protecting Anonymity Systems Using Cover Traffic

Anonymity systems have used cover traffic (also called 'dummy messages') as a protection method almost from the beginning. Notable examples include ISDN-Mixes [19], which use ISDN channels.

Cover traffic also appears in the earliest Internet-based designs, like PipeNet, where all clients transmit at the same rate [5]. The details were not specified, however. In the Web MIXes, each client always sends the same number of messages per time slice, inserting dummies if it has no data to send [3]. Tarzan uses a different approach, where cover traffic is generated in small sets of nodes [10].

Fu, Graham, Bettati and Zhao [11] and Levine, Reiter, Wang and Wright [14] show that even constant-rate cover traffic is vulnerable to traffic analysis under certain assumptions. Fu et al. perform experiments and find such vulnerability in practice. They explain it at the low level by subtle timer interrupt delays caused by packets with user data. They do not perform further experiments to check if other operating systems are vulnerable and how (and if) this problem appears on a system with high network load.

Levine et al. [14] assume that cover messages are added only by end nodes. Network losses or the attacker may then introduce holes in the traffic and the holes may then be used to correlate flows. Levine et al. introduce an algorithm called ’defensive dropping’. Intermediate nodes are instructed to drop certain messages and this changes the flow characteristic. They also propose a method to cross-correlate between output flows and input flows. The sequences to be compared are calculated by counting the number of received messages during fixed time intervals. We use the same method in this work and explain it in more detail in Section 6.1.5.

In a Freedom system security analysis, Back, Goldberg and Shostack [2] state that cover traffic was not included because of unresolved issues. Dingledine, Mathewson and Syverson express a similar opinion in the paper on Tor design, stating that the benefits are unclear while the costs are high, and await a design that will defend against a realistic adversary [9].

Zhu, Fu, Graham, Bettati and Zhao [27] study flow correlation attacks in traditional mix systems. They analyse different mix types using mutual information and frequency analysis to correlate flows, and show that existing designs are vulnerable. They propose an algorithm that inserts cover messages into all output flows of a node with exactly the same timing, but buffers received messages for a short time to limit the number of additional messages slightly.

Shmatikov and Wang [22] present an adaptive padding algorithm, where additional messages are inserted into statistically unlikely gaps in the flow. They use a traffic distribution from real-world traffic. As the authors note, the algorithm may provide less protection if the distribution is different from the assumed one. The authors mention the possibility of dynamically modifying the distribution, but they do not discuss this further.

Some studies have examined the methods the adversary may use. An interesting example is the work of Yu, Fu, Graham, Xuan and Zhao [26], who propose a signal watermarking technique based on Direct Sequence Spread Spectrum (DSSS) that uses a Pseudo-Noise (PN) code. The attacker adds a secret spread spectrum signal by slightly changing the sender’s traffic rate and recovers the signal after the flow is transmitted to the receiver.

2.2 Measuring Anonymity

The anonymity definition by Pfitzmann and Köhntopp we show in Section 1 defines anonymity using anonymity set. The metric based on anonymity set was used in previous studies, but it requires a method to decide if a certain node should be included in the set or not. For instance, the works of Kesdogan, Agrawal and Penz [12] and Kesdogan, Egner and Büschkes [13] define anonymity set in such a way that they include a node into the anonymity set if the probability of the desired role of the node (sender or receiver) is higher than zero. Their definition and the definition of anonymity set assumes that the decision of including a certain node in the anonymity set is always binary: yes or no. It may be, however, that different senders and receivers may send or receive messages with different probabilities. The anonymity set metric does not take this into account.

A popular anonymity metric that solves this problem has been independently introduced by Serjantov and Danezis [20] and Díaz, Seys, Claessens and Preenel [8]. It is calculated under an assumption of a specific attack. If, after an attack, an attacker can assign probabilities p_i to the senders and the number of senders is N, then we can calculate the entropy:

$$H(X) = -\sum_{i=1}^{N} p_i log_2(p_i). \tag{1}$$

It can be interpreted as the number of bits the attacker must learn to determine the identities of the senders or the receivers.

Díaz et al. introduced a metric using $H(X)$ and the maximum possible entropy of a set of N denoted as $H_N = log_2(N)$[1] introduced for normalisation. The metric is called the degree of anonymity and is defined as follows:

$$d = \frac{H(X)}{H_N}. \tag{2}$$

They also proposed an extension to their metric when the analysis must cover multiple cases and the probabilities differ in each of those cases. Their first proposal was to calculate a weighted sum of degrees of anonymity:

$$d = \sum_{j=1}^{K} p_j d_j, \tag{3}$$

where K is the number of cases, p_j the probability of certain circumstances and d_j is the degree of anonymity under these circumstances. The alternative method is to compare the calculated degrees with a certain assumed threshold and raise an alarm if the anonymity is below the threshold.

[1] The original work of Díaz et al. uses H_M for maximum entropy. We prefer to use H_N to denote the maximum entropy for N nodes.

This method is used in practise to evaluate anonymity system designs. For instance, Serjantov and Newman use this method to analyse the anonymity of timed pool mixes [21]. The same method is later used by Steinbrecher and Köpsell [23] to measure unlinkability.

The recent work of O'Connor [17] formally shows, using the entropy-based metric, how a message-based system using traditional mixes degrades over time: geometrically after certain threshold. The analysis is performed only for message-based systems.

The degree of anonymity was later analysed by Tóth, Hornák and Vajda [24] who highlighted situations where the calculated degree of anonymity is the same, but the protection given to a particular party varies greatly. They also provide examples of such cases. The simplest example has two anonymity systems: in the first one probability is the same for all participants. In the second system, however, the probability for the actual sender is 0.5 and for the others it is distributed equally. It turns out that we can find the number of nodes in both anonymity systems that the entropy is the same, for instance 20 nodes in the first anonymity systems and 101 nodes in the second one. Obviously the protection provided by those systems, as seen by the users, is not the same.

In this work we use the degree of anonymity metric to evaluate our algorithm. The problems highlighted by Tóth et al. are valid and should be taken into account when analysing the results, but at this time there is no other widely accepted anonymity metric for anonymity systems. Note also, that we use a weighted degree of anonymity that takes into account all possible cases.

3 Assumptions and Models

We assume that the anonymity system consists of a number of nodes that exchange fixed-length encrypted messages. As each node re-encrypts (and possibly modifies) the messages, an attacker cannot distinguish between messages based on the content. That leaves the inter-arrival times of the messages as a source of attacks.

The anonymity system transmits data in organised flows. Intermediate nodes can distinguish the flows in order to route them correctly. We do not cover the algorithm for route selection and establishment in this work: we assume that this phase has already been completed, and all flows have been successfully established. All messages have the same size.

Our attacker is very powerful. Not only can it monitor all the links between the nodes, but it also has the power to modify traffic by adding artificial delays to chosen messages. In our analysis, the adversary tries to match source and destination nodes by correlating the number of sent and received messages on selected links. All the algorithms used in the system and the structure of the network are known to the attacker, who can also distinguish between flows, even if they use the same link.

4 A Class-Based Cover Traffic Algorithm

4.1 Core Ideas

We design our algorithm to be used in an anonymity system where available bandwidth is limited and is a valuable resource. It also means that it may not be possible to fully protect every transmission in that anonymity system. There should be, however, a method to notify users of the current estimated protection level.

The algorithm presented in this work is based on four ideas: fixed time rounds for calculating the amount of cover traffic, different flow classification and protection methods for different classes, history information, and finally, the observation that certain flows are harder to protect than others.

4.1.1 Rounds

The basic unit of time is one round. During a round, a node receives and buffers messages, while sending messages from the previous round. At the end of each round, nodes add cover traffic messages to each flow. Then, scheduling takes place and a new round begins. The length of a round is a system-wide constant.

The existence of rounds introduces delays. The delay, if used with an admission control algorithm, is constant and may be changed at network configuration time.

The main reason for introducing rounds is the possibility of calculating the required amount of cover traffic from the real traffic statistics, and scheduling all messages equally during the next round. There are other reasons as well: for example, a slight change in inter-packet times during a round.

4.1.2 Flow Classification and Class-Based Protection

Flows are classified by their bandwidth and changes in bandwidth usage during subsequent rounds. When there are different traffic classes, each class may be protected in a different way. For instance, low-bandwidth classes may be protected more than high-bandwidth ones. The rationale for such behaviour is simple: less cover traffic is required to change all flows in a low-bandwidth class to fit the same (also low-bandwidth) pattern.

With more than one traffic result pattern, as in the traditional solutions, changing the flow to fit the closest pattern from a set of available ones should require fewer resources.

Introducing flow classification and class-based protection parameters has one drawback, however. Each class should contain roughly the same number of flows. Otherwise, the attacker may use the fact that the flow is one of only a few in its class.

4.1.3 Use of History

Our algorithm uses flow history rather than only using the flow class from the current round. This helps smooth rapid changes (like holes in the traffic) caused by, for instance, congestion. It also limits space for the attacker, as changes in flow parameters will be smoothed within the history length.

4.1.4 Different Levels of Protection for Different Flows

If there are major differences between the flows (for example, in the amount of bandwidth used), it may be more expensive to protect some flows than others.

Thus, we choose to differentiate protection. We offer better protection to lower-bandwidth flows, as this requires fewer additional messages. Such behaviour requires notifying users of the situation and, for instance, negotiating lower bandwidth if they require higher-grade anonymity.

4.2 The Basic Algorithm

Nodes process received messages during so-called rounds. Packets received during a round are buffered. When the round ends, for each flow we calculate the maximum number of packets received during a fixed number of previous rounds (the history is taken into account here), including the current round. If there are buffered messages from the previous rounds, we add their number to the maximum. Then we compare the value with the maximum bandwidth of the flow. If the maximum bandwidth is greater than or equal to the number of messages, then we classify the flow using the second value. On the other hand, if the number of messages received is greater than the maximum bandwidth, we classify the flow as its maximum bandwidth and put the last messages above the maximum threshold into the buffer. If the buffer fills up during that process, then we drop the remaining messages.

Let i be the current round and j be the current flow. $in[j][i]$ denotes the number of input messages for flow j during round i and $out[j][i]$ denotes the number of output messages for flow j during the round i. $B_{max}[j]$ is the maximum bandwidth for the flow j, and H_{len} is the history length. $M[j]$ is a temporary variable that stores the maximum number of messages for flow j received during the last H_{len} rounds. $class(N, j)$ is an algorithm that finds the class of the flow from the number of messages N for the flow j, and returns the total number of messages that should be sent during the next round.

The table $in[j][-H_{len}]..in[j][0]$ is initialised with $B_{max}[j]$ to hide the characteristics at the beginning of the flow and to protect very short flows.

We can then decide how many cover messages to insert using the algorithm 1 (with the details of handling messages above the limit omitted for clarity's sake).

An example of the $class()$ function is presented later in Section 6.

Algorithm 1. onRoundEnd(i,j), calculate the amount of cover traffic for the given round i and the flow j

```
Require: $i \geq 0, j \geq 0$
  $M[j] = max(in[j][i - H_{len}]..in[j][i-1])$
  if $M[j] \leq B_{max}[j]$ then
    $out[j][i] = class(M[j], j)$
  else
    $out[j][i] = class(B_{max}[j], j)$
  end if
```

4.3 Extensions

Numerous extensions to the basic algorithm are possible. For instance, we may want to increase the length of the flow of more than H_{len} rounds beyond what the basic algorithm does. Such an extension would let us hide the flow's end and possible short-term gaps. Additionally, the increase of length should not be easily predictable. For such cases, we propose an extension that alters the flow based on the rate at which the flow uses allowed and allocated bandwidth. Flows using less bandwidth have their flow length increased more.

In addition to the variables used in the basic version of the algorithm, $fill\_class(f, j)$ is a function that uses the flow parameters and the ratio of used bandwidth f for the flow j to return the base of the increase in flow length. The table $fw[j]$ stores the base increase for the flow j and $flow\_wait[j]$ stores the number of rounds that remain for the flow j. The calculation of the increase additionally uses the default increase D, which is a system constant. $rand(X)$ returns a random integer value in the range of $[0, X)$.

Algorithm 2. onRoundEnd(i,j), calculate the amount of cover traffic for the given round i and the flow j (extended version of algorithm 1)

```
Require: $i \geq 0, j \geq 0$
  $M[j] = max(in[j][i - H_{len}]..in[j][i-1])$
  $S = 0$
  for $k = (i - H_{len})$ to $(i-1)$ do
    $S = S + in[j][k]$
  end for
  $f = (S * 100)/(H_{len} * M[j])$
  if $M[j] \leq B_{max}[j]$ then
    $out[j][i] = class(M[j])$
  else
    $out[j][i] = class(B_{max}[j])$
    $fw[j] = fill\_class(f, j)$
  end if
```

Algorithm 3. onCoverGeneration(i,j), add additional cover traffic to hide the temporary lack of traffic for the given round i and the flow j

```
Require: $i \geq 0, j \geq 0$
  $c = fw[j] - out[j][i]$
  if $c \leq 0$ then
    return
  end if
  if $out[j][i] = 0$ then
    if $flow\_wait[j] > 0$ then
      $flow\_wait[j] = flow\_wait[j] - 1$
    else
      $flow\_wait[j] = 0$
    end if
  else
    $flow\_wait[j] = rand(fw[j]) + D/2 + 1$
  end if
  $out[j][i] = out[j][i] + c$
```

The pseudo-code shown in algorithms 2 and 3 describes our algorithm (with the details of handling messages above the limit omitted for clarity's sake, as in the previous version).

An example of the $fill\_class()$ function is presented later in Section 6.

4.4 Discussion

The idea of rounds is similar to the mix time-out from timed mixes [7], but the details and purpose are different. During a round, messages between two nodes are transmitted in the same order in which they were received, while mixes may change that order. The purpose of introducing rounds is to add the option of adding cover traffic messages at equal intervals between the data messages. This would not be possible without knowing the number of messages received during the round. We introduce rounds as a feature that enables protection, not as a protection mechanism itself.

Uniform distribution of packets during rounds and scheduling is an essential part of the proposed solution. It destroys the low-level packet inter-arrival time distribution, but should not introduce additional artifacts. It is also important to take into account that observers will not be synchronised with the round clock, so they may get different counts of packets during rounds. When doing tests, we had an implementation that did not use the uniform distribution and simply scheduled cover messages at the beginning of the round. Especially for longer flows, it generated artifacts that were sufficient for the attacker to find a good correlation.

The mechanism of history and flow classification based on the maximum value in the history window limits the number of events of bandwidth change, as the attacker may use such events to correlate flows. The same mechanism allows fast recovery of high use of bandwidth.

It should be noted that the algorithm is designed to limit the number of cover traffic messages in common situations. In the worst case, if the attacker sends B_{max} during one round every H_{len} messages, the algorithm will work just as the full cover traffic. This situation may be detected however, using only limited additional resources, as the history is available for each node.

The algorithm does not hide the connection start. The cover traffic mechanism does not seem to be the right place for that protection as one of our requirements is low delay. Another mechanism should be used to prevent the attacker from using the timings of flow starts. It may be performed for instance by predicting user activity and starting a new connection earlier.

In the above pseudo-code, we assume that the number of messages received during a round does not exceed the maximum bandwidth. We made this simplification in order to keep the description and pseudo-code clear. In real implementation, it would be easy to add buffering of additional messages. However, the implementation details will depend on the transport protocol used, as it determines whether messages may (or should be) dropped and whether flow control algorithms should be used (and if so, which ones).

5 Theoretical Evaluation

In this section we present a theoretical evaluation of our algorithm using the degree of anonymity metric explained in Section 2.2. Both simple and extended versions of the cover traffic algorithm are modelled here, but the later one with the assumption of the same extension given to all traffic classes.

5.1 Modelling the Anonymity System

We assume an anonymity system with N senders and N receivers. The input and output links of the anonymity system are observed by the attacker.

The possible observed values come from the alphabet $\mathscr{A}$. The alphabet may be interpreted differently depending on the attacker's capabilities. For instance, the attacker may only have information if any messages occurred in the given time interval. Then $\mathscr{A} = \{0,1\}$. On the other hand, the alphabet may be directly interpreted as the number of messages observed, or as an interval identifier.

Observations from subsequent time intervals form sequences. S_i denotes the sequence observed for the node i, $S_i = \{s_1, s_2, ..., s_L\}, s_i \in \mathscr{A}$. The observed sequences have the length L. Let us denote $\mathscr{S}$ as the set of all possible sequences of length L.

Each of the senders I_i outputs one sequence S_i. Those sequences are then processed by the anonymity system and result in output sequences R_j received by the receivers O_i.

5.1.1 Assigning Probabilities

We denote $P(S_i)$ as the probability that the input sequence is S_i. $P(R_j|S_i)$ denotes the probability of the output (received) sequences R_j if the sent sequence was S_i.

Our formulas include the probability of each sequence $P(S_i)$, so it can be used with any traffic distribution. However, for comparisons between different cover traffic algorithms, we choose to use a traffic distribution that produces every sequence with equal probability. That allows us to remove the effects caused by the assumptions about the traffic distribution and analyse all cases, even those that would have a very low probability in practice, but may be used by the attacker.

Given the traffic distribution that produces our inputs and the model of the anonymity system, for each observable input sequence we can calculate all possible outputs with their probabilities:

$$M_i = \{S_i, P(S_i), \{\langle R_0, P(R_0|S_i)\rangle, \langle R_1, P(R_1|S_i)\rangle, ..., \langle R_N, P(R_N|S_i)\rangle\}\}. \qquad (4)$$

With the tuples calculated for each input sequence S_i we can calculate the receiver and sender anonymity, the number different output sequences and the number of receivers with the same sequence.

Iterating all possible input sequences S_i we can calculate the receiver anonymity from the formulas (1) and (2).

For the analysis, the number of alphabet symbols is reduced to 2 (except for the analysis of the dependency on the alphabet size). The following parameters are taken into account: sequence length L, history length H_{len}, the alphabet size and input traffic distribution. For the extended version of the algorithm the list also includes the maximum extension length.

As the calculations are computationally intensive, they have been performed for the ranges of parameters where the results are expected in acceptable time.

5.1.2 Finding Output Classes

We can get the number of output classes directly by searching for all observable R_j. The probability that a node belongs to a specific class can be calculated as follows:

$$P(R_j) = \sum_{i=1}^{N} P(R_j|S_i) * P(S_i). \qquad (5)$$

We can also calculate the number of elements in each class R_i given the number of inputs and their sequences.

5.1.3 Calculating the Degree of Anonymity

Let us assume, that the attacker knows the sender and wants to uncover the receiver. The attacker knows the input traffic distribution and all the anonymity system algorithms, so one can assign probabilities $P(R_j|S^*)$ of the anonymity system changing the input sequence S^* to all possible output sequences R_j.

Output nodes can then be divided into groups based on their R_j. Each of the receivers in one group may be the addressee of the sequence the attacker is tracking, and thus the attacker will be unable to distinguish between the nodes in a group without further information. On the other hand, we are able to calculate the probability of each output sequence R_j using the equation (5). It means that we can also calculate the expected size of each R_j group. The expected size of the anonymity set of R_j is then:

$$E_{R_j} = N * P(R_j) = N * \sum_{k=1}^{N} P(R_j|S_k) * P(S_k). \tag{6}$$

Iterating all possible input sequences S_i we can calculate the receiver anonymity from the formulas (1) and (2). The only problem is the number of nodes in each group, as the expected values of E_{R_j} may not be integers. In this work we round the number of nodes in the set $F_{R_j} = round(E_{R_j})$ and assume that the situation we consider is an equivalent to F_{R_j} nodes, each with probability $P(R_j|S_i)/F_{R_j}$ of being the receiver, for $F_{R_j} > 0$. Alternatively, we can calculate the anonymity for the set by either rounding up or down. If E_{R_j} is greater than 0, but less than 1, we assume $F_{R_j} = 1$, as we do not want to ignore the classes of low probability:

$$F_{R_j} = \begin{cases} round(E_{R_j}) & \text{for } E_{R_j} \geq 1. \\ 1 & \text{for } 0 < E_{R_j} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

However, there is also a need to consider the effect of rounding the size set, as the total number of considered outputs may be higher than N. Rounding changes the input traffic distribution and the number of considered nodes. The increased number of nodes is denoted as M. It is the sum of all expected numbers of nodes over all possible output classes:

$$M = \sum_{i=1}^{N} F_{R_i}. \tag{8}$$

The degree of anonymity d_r may then be calculated as:

$$\begin{aligned} H_i(X) &= -\sum_{j=1}^{N} \sum_{k=1}^{F_{R_j}} \frac{P(R_j|S_i)}{F_{R_j}} * log_2 \frac{P(R_j|S_i)}{F_{R_j}} \\ &= -\sum_{j=1}^{N} F_{R_j} \frac{P(R_j|S_i)}{F_{R_j}} * log_2 \frac{P(R_j|S_i)}{F_{R_j}} \\ &= -\sum_{j=1}^{N} P(R_j|S_i) log_2 \frac{P(R_j|S_i)}{F_{R_j}}, \end{aligned} \tag{9}$$

$$d_r = -\sum_{i=1}^{N} P(R_j) \frac{H_i(X)}{log_2 M}. \tag{10}$$

5.2 Basic Version

In the simple version, the algorithm uses only the values from its history and the output is equal to the maximum value in the history window. Therefore, the protection depends on the history length H_{len} and the alphabet size. Generally, we expect that the higher H_{len} is, the higher the protection will be. When it comes to the alphabet, the less symbols, the less different possibilities for the attacker, so the protection should be higher.

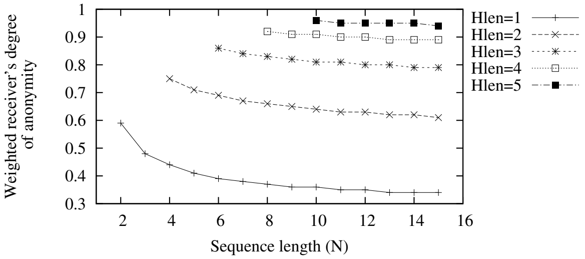


Fig. 1 Receiver's degree of anonymity as a function of sequence length for the basic version of our cover traffic algorithm for different history window sizes H_{len}

Figure 1 shows the degree of anonymity as a function of sequence length for history length from 1 to 5. The values for sequence length less than twice the H_{len} were omitted, as it is heavily dependant on initialisation. It may be seen that increased H_{len} does indeed increase protection and even for relatively short $H_{len} = 2$ the degree of anonymity raises to acceptable 0.7 for long sequences. Additional increases of the value, however, have less and less impact with the increase of H_{len}. This effect

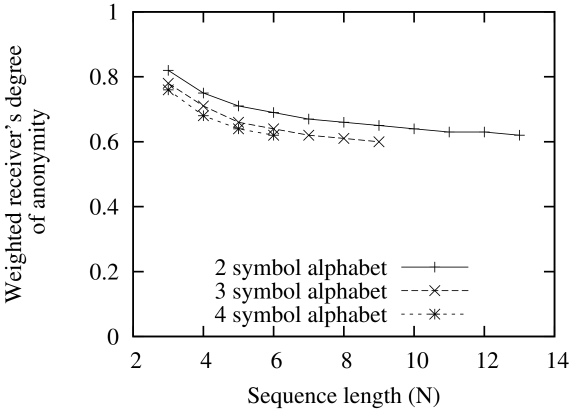


Fig. 2 Receiver's degree of anonymity as a function of sequence length for the basic version of our cover traffic algorithm for different alphabet sizes, $H_{len} = 2$

may be explained by the fact that the number of 'gaps' to be filled becomes smaller and the shorter 'gaps' in the traffic have been already filled at lower H_{len}.

The more symbols in the alphabet, the more possible sequences and also the less protection. The degree of anonymity for 2, 3 and 4-symbol alphabet is shown in Fig. 2. The differences are visible, but small. For the same ratio, the higher number of symbols means much higher number of nodes in rare groups. However, it should be noted, that the number of possible symbols is different as it depends on the alphabet size, so it is 2^i for 2-element alphabet, 3^i for 3-element alphabet and so on.

5.3 Extended Version

The extended version of the algorithm is very similar to the basic one, but includes one modification. The flow may be extended further after H_{len}, for the maximum of D rounds. The length of the extension is an integer value in the range $[1, D)$ with equal probabilities. We denote the situation with no extension as $D = 0$.

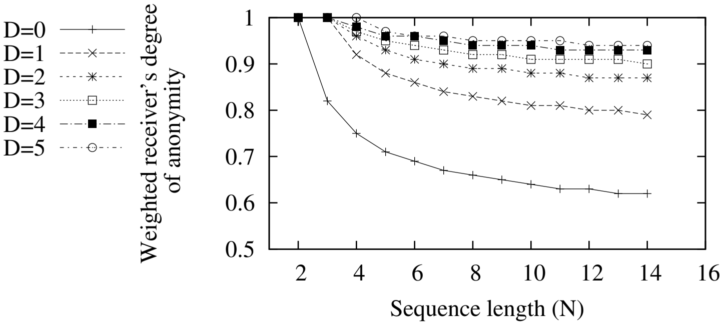


Fig. 3 Receiver's degree of anonymity for different values of maximum extension D, extended version of the algorithm, $H_{len} = 2$

Figure 3 shows the receiver's degree of anonymity for different values of the extension D. The increase of D changes the degree of anonymity in a similar way as H_{len} (compare Fig. 1). The difference is that the increase of H_{len} and D change the number of additional messages to be sent (so the cover traffic cost) in slightly different ways.

The conclusion is that the extended version may increase the protection for a common case, but it introduces artifacts that may be used by the attacker.

5.4 The Influence of Input Traffic Distribution

The input traffic distribution is a parameter of our model, but it is an important factor at the implementation phase. That is why we compare the results for the uniform distribution with a different distribution. The distribution chosen models an ON-OFF source, which may be understood as if there were two main states: sending (ON, 1) or not sending (OFF, 0). This model is relatively simple and widely used in the literature [1, 4] and available in network simulators [15]. The length of its stay in each of the states comes from the exponential distribution, each with independent λ parameter. Figure 4 shows receiver's degree of anonymity for different input traffic distribution: uniform like in the previous examples and different versions of the ON-OFF distribution for three values of each of the two lambda parameters of the exponential distributions. It suggests that the algorithm protects ON-OFF traffic better than traffic of uniform distribution. This is related to the assumptions the algorithm was designed under and the mechanism of extending flows (that the flows are visible n the traffic). The differences between the different lambda parameters are small.

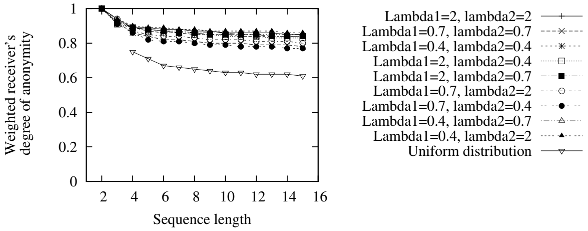


Fig. 4 Receiver's degree of anonymity as a function of sequence length for the basic version of our cover traffic algorithm for different input traffic distributions, $H_{len} = 2$. Lambda1 is used for in the state of non-sending and lambda2 is used in the state of sending.

6 Practical Evaluation

We implemented the proposed algorithm and compared it with other algorithms from the literature in terms of increased protection and cost (the amount of additional traffic).

6.1 Implementation and Test Configuration

For evaluation, we have implemented and tested our cover traffic algorithms using the ns-2 simulator [16]. We have compared our algorithm with a situation without any cover traffic, with the algorithm of Zhu et al. and with the algorithm of Shmatikov and Wang. It should be noted that we had to modify the two last algorithms to work properly in our test configuration. We present the modifications later in this section.

6.1.1 Simulated Network Configuration

The network configuration consisted of 320 source nodes, 320 destination nodes and a single intermediate node. Each of the source nodes transmitted a single flow to one of the destination nodes. There were the following types of traffic:

- 64 constant rate flows at rates from 32 to 256 kbit/sec (class 1)
- 64 constant rate flows with random intervals between the packets, at rates from 32 to 256 kbit/sec (class 2)
- 64 flows using a Pareto distribution of delay between the packets, at rates from 32 to 256 kbit/sec (class 3)
- 64 flows using exponential distribution of delay between the packets, at rates from 32 to 256 kbit/sec (class 4)
- 16 flows with bursts, burst length and interval of 0.1, 0.2, 0.5 or 1.0 second at 256 kbit/sec (class 5)
- 16 flows with single burst of length of 10, 20, 30 or 40 per cent of the simulation time at 256 kbit/sec(class 6)
- 32 PN (pseudo-noise) sequences with bit length of 0.2, 0.5, 1.0 and 2.0 seconds like proposed in [26] at 256 kbit/sec (class 7)

For each of the five algorithms, the simulation times ranged from 5 to 300 seconds (5, 10, 20, 30, 40, 50, 75, 100, 150, 200, 250 and 300 seconds) and each simulation was repeated 11 times. All packets had data fields of 512 bytes.

Each flow was bi-directional. One of the directions transmitted data, while the other remained silent. If the cover traffic algorithm transmits data on all open flows (which is true for our algorithm and for the algorithm of Zhu et al.), cover traffic was also generated on the silent ones and included in the further cost calculations. The silent flows were not, however, directly taken into account when calculating protection ratio.

Classes 1 to 4 represent different distributions of traffic, while classes 5 to 7 show different methods of possible attacks, where the attacker modifies the traffic to form a known pattern. We use different distributions of traffic, because there is not enough data on the flow characteristic of deployed anonymity systems. Additionally, we assume that different protection of different traffic classes and possible traffic shaping (which we suggest) may change user behaviour patterns, so also the traffic distribution. Flows that differ from the others may be also introduced by the attacker.

6.1.2 Our Algorithm

While testing our algorithm, we used two sets of parameters, each with a round time equal to 0.1 sec.

The first one, 'Classes, $D = 5$' used used four classes for the $class()$ function with $H_{len} = 16$ (in rounds, that equals 1.6 seconds), $B_{max}[i] = 7$ for each i (7 packets/sec, approx 280 kbit/sec) and four classes for the $fill\_class()$ function with $D = 5$ rounds (equalling 0.5 seconds).

The $class()$ and $fill\_class()$ functions used in the simulations may be presented with the pseudo-code shown in algorithms 4 and 5, respectively.

Algorithm 4. class(N,j), return the amount of cover traffic for the flow j after classifying the number of messages N

```
**if** $N/B_{max}[j] > 0.75$ **then**
    **return** $B_{max}[j]$
**end if**
**if** $N/B_{max}[j] > 0.5$ **then**
    **return** $3 * B_{max}[j]/4$
**end if**
**if** $N/B_{max}[j] > 0.25$ **then**
    **return** $B_{max}[j]/2$
**end if**
**return** $B_{max}[j]/4$
```

Algorithm 5. fill_class(f, j), return the maximum extension of the flow f for round j

```
**if** $f > 0.75$ **then**
    **return** $D * 2 + 1$
**end if**
**if** $f > 0.5$ **then**
    **return** $D + 1$
**end if**
**if** $f > 0.25$ **then**
    **return** $D/2 + 1$
**end if**
**return** $D/4 + 1$
```

Finally, the second variant, ’Classes, always cover’ used the same four classes and parameters for the $class()$ and $fill\_class()$ functions as the previous one. However, the $onCoverGeneration()$ function is changed. If the number of received messages is lower than the current increase base $fw[j]$ for the flow j, it sends cover messages up to $fw[j]$. This can be presented with the pseudo-code from algorithm 6.

Algorithm 6. onCoverGeneration(i,j), add additional cover traffic to hide the temporary lack of traffic for the given round i and the flow j

```
$c = fw[j] - out[j][i]$
**if** $c \leq 0$ **then**
    **return**
**end if**
$out[j][i] = fw[j]$
```

6.1.3 The Algorithm of Zhu et al.

Zhu et al. [27] define the algorithm only for two sources and two destinations. We have extended it for more nodes so as to generate cover messages for all connections at the same time if they do not have any messages in the queue. If there is a message, it is sent. The authors suggested an extension by sending only on a limited number of connections, but did not provide an algorithm. We consider the choice a complicated problem, so we leave the algorithm in its base form.

6.1.4 The Algorithm of Shmatikov and Wang

The algorithm of Shmatikov and Wang [22] uses a traffic distribution with an average rate much higher than that in our simulation. Because of that, we have scaled the distribution so as to multiply each delay by ten. That yields an acceptable average inter-packet delay.

6.1.5 Result Processing

During the result analysis phase, we tried to correlate input and output flows. We calculated cross-correlations between the number of packets in each 0.1-second interval between the input and output links from each single simulation. We used the following equation:

$$r(d) = \frac{\sum_i [(x_i - mx) * (y_{i-d} - my)]}{\sqrt{\sum_i (x_i - mx)^2}\sqrt{\sum_i (y_{i-d} - my)^2}}, \tag{11}$$

where d is delay, r is cross–correlation, x_i and y_i are the signals, $i = 0, 1, ..N - 1$. mx and my are the mean values of the x and y signals, respectively.

Using the maximum $r(d)$ from each input-output pair, we calculated the best match for each input. Then we checked whether the best match was correct. We later refer to the ratio of correct matches as the detection rate.

6.2 Results

Figure 5 shows the rate of incorrect matches (source and destination not matched) for different algorithms as a function of flow length. The curve has a very similar shape for most of the algorithms: the detection rate is low for the shortest flow, then increases rapidly until the flow length reaches approximately 40 seconds. Then the rate still increases, but at a slower pace.

Unsurprisingly, the algorithm of Zhu et al. [27] gives the best protection, as it transforms each of the input flows into the same output. Full cover traffic with rounds yields similar results.

Of the other algorithms, the ones proposed in this work provide decent protection, above 70 per cent or above 40 per cent depending on the parameters, for the

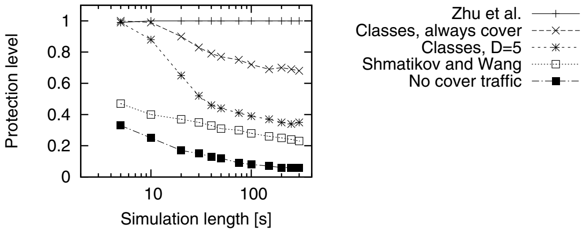


Fig. 5 Protection given by each of the simulated algorithms depending on the flow type, for all flows

longest flows. It should be noted, however, that there is a strong dependency on the simulation time. The shorter the simulation, the better protection is gained.

Surprisingly, the algorithm of Shmatikov and Wang did only a little better than the case with no protection at all. On investigation, we found out that it inherits too many of the original characteristics if the traffic distribution is not very similar to the assumed one.

6.2.1 Passive and Active Attacks

Figures 6, 7, 8 show the average protection rates for different traffic classes. The algorithm that transforms all flows to the same form shows similar results independently of the flow type.

The other algorithms clearly work better against passive attackers as shown in Fig. 6. The protection against active attackers is lower as shown in Fig. 7 and 8. It should be noted that the protection given by the algorithm of Shmatikov and Wang is very low under active attacks. Our algorithm provides significant protection for short observation lengths, up to approximately 50 sec (500 rounds).

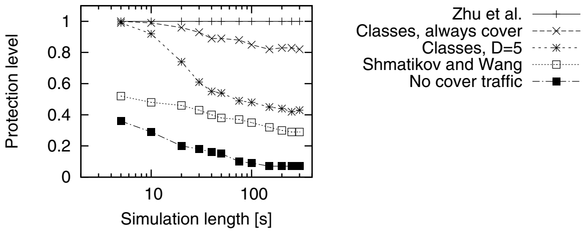


Fig. 6 Protection given by each of the simulated algorithms under passive attack conditions (flow types 1–4)

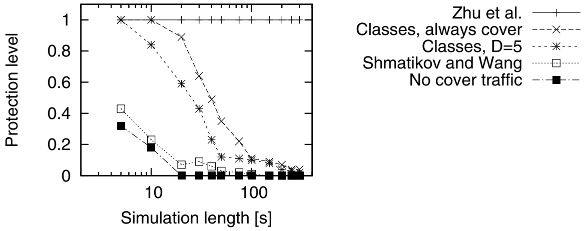


Fig. 7 Protection given by each of the simulated algorithms for flows with a single burst of traffic (flow type 6)

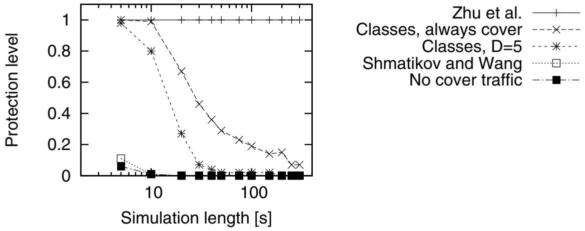


Fig. 8 Protection given by each of the simulated algorithms for flows with PN sequences (flow type 7)

6.2.2 Protection Cost

Detection ratio is an important factor when evaluating cover traffic algorithms, but it does not include all of the important aspects. The most important of the remaining ones is probably the cost of the algorithm, defined as the number of cover messages it must send.Table 1 shows the cover traffic ratio for different simulation lengths.

The higher cover ratio of our algorithm on short flows is caused by the initialisation of $in[j][i]$ to $B_{max}[j]$ (see 4.2). Other than that, values do not change significantly with simulation time. We can see that our algorithms provide significant reduction compared with Zhu and small reduction compared with Shmatikov and Wang. An ideal algorithm would have few cover messages and a low detection ratio (high

Table 1 Cover traffic ratio for different simulation lengths

Algorithm	Cover traffic ratio			
	T=5	T=20	T=100	T=300
Classes, $D = 5$ rounds	0.42	0.37	0.37	0.36
Classes, always cover	0.49	0.45	0.44	0.44
Shmatikov and Wang	0.45	0.45	0.45	0.45
Zhu et al.	0.60	0.60	0.60	0.60

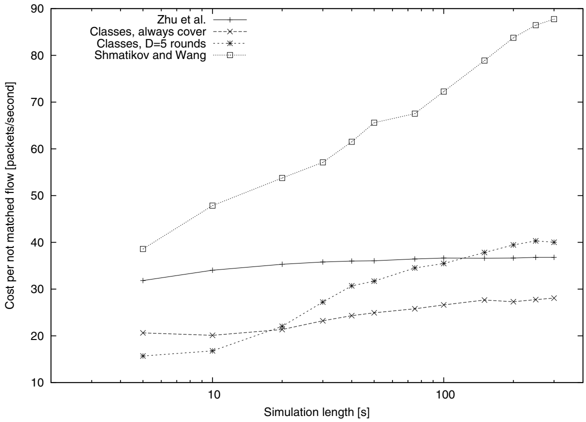


Fig. 9 Cost per each undetected flow for each of the simulated algorithms

protection). We have not found a comparison metric that incorporates both these aspects. Because of this, we propose a new metric, which we present below.

Using the same input traffic, we may compare different algorithms using the number of cover messages generated during a fixed interval T, divided by the number of successfully protected flows $N_{protected}$, where we define $N_{protected}$ as the number of flows not correlated by the attacker under certain attack scenerio. This can be presented as:

$$cost = \frac{\sum_{t=0}^{T_{total}-T} cover\_between(t,t+T)}{N_{protected} * T}. \tag{12}$$

Figure 9 shows the cost, as defined in (12), $cost_{T=1sec}$. It is worth noting that there are algorithms that scale well and those where the cost increases significantly when the simulation becomes longer.

Our algorithm has the lowest cost, but the set of parameters changes. For short flows, a small D is enough, but the cost of the algorithm with $D = 5$ increases, and even becomes higher than the cost of the algorithm of Zhu et al., for flows slightly longer than 100 seconds. For flows longer than 20 seconds, the second version of our algorithm has the lowest cost per undetected flow and has significantly lower cost than the algorithm of Zhu et al.

Shmatikov and Wang's algorithm in our simulations has a cost even higher than the algorithm of Zhu et al., and this cost increases quickly.

7 Conclusions and Future Work

We have shown that good protection against passive and short active attacks is possible with lower cost than in previous algorithms. Our class of algorithms addresses the basic conflict between protection and performance in the design of anonymity systems. Our simulations show that our algorithm uses less cover traffic than the algorithms that send the same traffic on all flows (36 to 44 per cent of the traffic for long simulations compared to 60 per cent needed by the algorithm of Zhu et al.). At the same time, the protection level is higher than in the algorithm of Shmatikov and Wang (40 to above 70 per cent compared to 20 per cent for long simulations).

Our algorithm provides decent protection against passive attacks. It also increases the time needed to perform an active attack. Further protection against active attacks may try to actively detect such attacks and perform countermeasures. The algorithms for such detection and the possible actions to perform are left for further study.

We have shown that our algorithm protects well against passive attacks, but less well against active attacks, especially long-lasting ones. However, our method leaves room for active attack detection. This can be done if the nodes transmit flow statistics to the initiator. These statistics would include the number of data and cover messages in each round. Based on those data, it should be possible to detect suspicious traffic patterns. That would include a situation when attackers have control over the first link and introduce delays.

The algorithm should be used by all intermediate nodes in the anonymity system. Then the traffic will be at least partially protected if the attackers introduce delay on certain links or have control over one of the intermediate nodes and those nodes fail to follow the prescribed cover traffic scheme.

We do not assume a specific transport protocol used in the anonymity system, and we do not address behaviour on losses. However, the protocol is optimised for datagram-based transport, with retransmissions handled (if necessary) at the higher (application) level. This is because we introduce strict restrictions on the flow. If used with a connection-based protocol (like TCP), the algorithm requires an implementation of flow control on the whole path.

Another parameter worth mentioning is the number of traffic classes. The number and behaviour of classes should be set based on the traffic in the specific anonymity system. Different extensions are possible here, keeping in mind that the more classes there are, the fewer flows there are in a single class, and the anonymity set becomes smaller. This requires planning, which may be done at the negotiation phase. Some balancing between classes may be desirable.

Our algorithm introduces latency. However, the latency is predictable. Additionally, we suggest the round time of 100 ms, what gives the total latency of 200 ms per hop. If that is too high, the round time may be shortened to 50 ms (100 ms per hop), or even further. We believe that such delay is acceptable, especially when the existing anonymity systems are reported to introduce latency measured in seconds [25]. Additional optimisations are possible, like processing the messages (especially encrypting/decrypting) during the latency period.

Finally, we also need new methods for evaluating anonymity in flow-based anonymity systems. Apart from the problems pointed by Tóth et al., the degree of anonymity metric which we use in this work also turned out to be complicated and time-consuming to calculate.

References

1. Adas, A.: Traffic Models in Broadband Networks. IEEE Communications Magazine 35(7), 82–89 (1997)
2. Back, A., Goldberg, I., Shostack, A.: Freedom Systems 2.1 Security Issues and Analysis. White paper, Zero Knowledge Systems, Inc. (May 2001)
3. Berthold, O., Federrath, H., Köpsell, S.: Web MIXes: A system for anonymous and unobservable Internet access. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 115–129. Springer, Heidelberg (2001)
4. Chen, T.: Characterization of ATM on-off traffic from cell traffic measurement. In: 3rd IEEE International Workshop on Broadband Switching Systems (BSS 1999), pp. 53–57 (1999)
5. Dai, W.: PipeNet 1.1. Post to Cypherpunks mailing list (November 1998)
6. Danezis, G., Diaz, C.: A Survey of Anonymous Communication Channels. Technical Report MSR-TR-2008-35, Microsoft Research (January 2008)
7. Diaz, C., Preneel, B.: Taxonomy of Mixes and Dummy Traffic. In: Proceedings of I-NetSec 2004: 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems, Toulouse, France, pp. 215–230 (August 2004)
8. Diaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 54–68. Springer, Heidelberg (2003)
9. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The Second-Generation Onion Router. In: Proceedings of the 13th USENIX Security Symposium, pp. 303–320 (August 2004)
10. Freedman, M.J., Morris, R.: Tarzan: A Peer-to-Peer Anonymizing Network Layer. In: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington, DC, pp. 193–206 (November 2002)
11. Fu, X., Graham, B., Bettati, R., Zhao, W.: On Effectiveness of Link Padding for Statistical Traffic Analysis Attacks. In: ICDCS 2003: Proceedings of the 23rd International Conference on Distributed Computing Systems, Washington, DC, USA, p. 340. IEEE Computer Society, Los Alamitos (2003)
12. Kesdogan, D., Agrawal, D., Penz, S.: Limits of Anonymity in Open Environments. In: Petitcolas, F. (ed.) IH 2002. LNCS, vol. 2578, pp. 53–69. Springer, Heidelberg (2003)
13. Kesdogan, D., Egner, J., Büschkes, R.: Stop-and-Go MIXes: Providing Probabilistic Anonymity in an Open System. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 83–98. Springer, Heidelberg (1998)
14. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing Attacks in Low-Latency Mix-Based Systems. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 251–265. Springer, Heidelberg (2004)
15. ns 2 manual. Exponential on/off, http://www.isi.edu/nsnam/ns/doc/node505.html (accessed May 2009)
16. ns-2 webpage, http://nsnam.isi.edu/nsnam/index.php/User_Information (accessed: January 2009)
17. O'Connor, L.: Entropy Bounds for Traffic Confirmation. Technical Report 2008/365, IACR (October 2008)
18. Pfitzmann, A., Köhntopp, M.: Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 1–9. Springer, Heidelberg (2001), http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.6.pdf

19. Pfitzmann, A., Pfitzmann, B., Waidner, M.: ISDN-mixes: Untraceable communication with very small bandwidth overhead. In: Proceedings of the GI/ITG Conference on Communication in Distributed Systems, pp. 451–463 (February 1991)
20. Serjantov, A., Danezis, G.: Towards an Information Theoretic Metric for Anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 41–53. Springer, Heidelberg (2003)
21. Serjantov, A., Newman, R.E.: On the Anonymity of Timed Pool Mixes. In: Proceedings of the Workshop on Privacy and Anonymity Issues in Networked and Distributed Systems, Athens, Greece, pp. 427–434. Kluwer Academic Publishers, Dordrecht (2003)
22. Shmatikov, V., Wang, M.-H.: Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 18–33. Springer, Heidelberg (2006)
23. Steinbrecher, S., Köpsell, S.: Modelling Unlinkability. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 32–47. Springer, Heidelberg (2003)
24. Tóth, G., Hornák, Z., Vajda, F.: Measuring Anonymity Revisited. In: Liimatainen, S., Virtanen, T. (eds.) Proceedings of the Ninth Nordic Workshop on Secure IT Systems, Espoo, Finland, pp. 85–90 (November 2004)
25. Wendolsky, R., Herrmann, D., Federrath, H.: Performance Comparision of the low-latency Anonymisation Services from User Perspective. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 233–253. Springer, Heidelberg (2007)
26. Yu, W., Fu, X., Graham, S., Xuan, D., Zhao, W.: DSSS-Based Flow Marking Technique for Invisible Traceback. In: SP 2007: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 18–32. IEEE Computer Society, Los Alamitos (2007)
27. Zhu, Y., Fu, X., Graham, B., Bettati, R., Zhao, W.: On Flow Correlation Attacks and Countermeasures in Mix Networks. In: Martin, D., Serjantov, A. (eds.) PET 2004. LNCS, vol. 3424, pp. 207–225. Springer, Heidelberg (2005)

Fault Recovery Performance Analysis of Functionally Distributed Transport Networking System

Kentaro Ogawa, Kenichi Higuchi, and Shinichiro Chaki

Abstract. We propose a fault recovery method in functionally distributed transport networking that separates the control-plane processing part (control element, CE) from the forwarding-plane processing part (forwarding element, FE) of the router. In this architecture, one path-control process in the CE consolidates and processes the path computations and the path settings for multiple FEs. This leads to reduction in the path-control complexity and efficient operation of large scale networks. On the other hand, it is absolutely critical to ensure the high reliability of the CE. We analyze the performance of the proposed fault recovery method by using software implementation.

Keywords: fault recovery; performance analysis; functionally distributed transport networking; redundant configuration; router.

1 Introduction

Traffic in IP networks has increased and services have diversified over the last few years. IP networks need to have scalability and high-quality data transmissions, and they must be able to incorporate new functions. The progress of large-scale IP networks has increased the complexity of the path control and has required the use of increasingly sophisticated routers. For instance, a high-performance processor and a huge memory are now required for the edge router

Kentaro Ogawa · Kenichi Higuchi · Shinichiro Chaki
Network Service Systems Laboratories,
NTT Corporation,
3-9-11, Midori-cho, Musashino-shi,
Tokyo, 180-8585 Japan
e-mail: {ogawa.kentaro,higuchi.kenichi,
chaki.shinichiro}@lab.ntt.co.jp
http://www.ntt.co.jp/islab/e/org/ns.html

S. Caballé, F. Xhafa & A. Abraham (Eds.): INCoS 2009, SCI 329, pp. 281–295.
springerlink.com

that supports VPNs and multicasts because routing protocols, such as open shortest path first (OSPF), border gateway protocol (BGP), and protocol independent multicast sparse mode (PIM-SM), operate at the same time. The interior gateway protocol (IGP) used for path control manages the topology of the network according to the link state database (LSDB) and computes the paths. An increase in the number of routers in a network enlarges the LSDB and lengthens the path computational time, which limits the scale of a network. In large scale network composed of hundreds of thousands of routers, the routing information is consolidated based on network domains within a defined area, and coordination between exterior gateway protocols (EGP) and IGPs for path control is needed. The IGP controls the path in the network domains, and the EGP controls the path between the network domains. As a result, routers must be set according to the network design, which increases maintenance operations. Moreover, the addition of functions to the transport stratum entails the addition of functions to an individual router. Because large scale networks have many and various routers, it takes a long time to provide new network services.

To overcome these problems, the softrouter architecture, with a control-plane processing server for high processing performance and ease of adding functions to the transport stratum, has been proposed [1]. This architecture separates the control-plane processing part (control element, CE) from the forwarding-plane processing part (forwarding element, FE) of the router. In this architecture, a CE is mounted on a general-purpose server platform, which becomes a control-plane-processing server, and one path-control process in the CE consolidates and processes the path computation and path setting for one or more routers, which act as FEs. The FEs are bound to an arbitrary server via an IP session. An FE collects topology information from adjacent FEs and sends it to a CE. The CE computes the path and sets up a routing table for the connected FEs. The performance and functions of the CE can be easily upgraded in this architecture. However, the softrouter cannot solve the complexity of the path control due to the coordination between IGPs and EGPs.

To reduce the complexity of the path control in large scale networks, we have proposed the functionally distributed transport networking architecture [2, 3] by expanding the softrouter concept. This architecture configures a functionally distributed router (FDR) with one CE and multiple FEs as well as the softrouter, and recursively configures the entire network with a higher CE and all of the FDRs. The higher CE treats each FDR as one router and computes the path for the entire network by using the same path-control method as one for the FDR. This leads to reduction in the path-control complexity and efficient operation of large scale networks.

As for a related work, OpenFlow [4] adopts the concept of the separation between the control-plane and the forwarding-plane. OpenFlow can also control multiple forwarding elements, called OpenFlow Switch, by a controller. However, OpenFlow needs the specific hardware for an OpenFlow Switch and the specific OpenFlow protocol for the communication between a controller and OpenFlow Switches.

The above studies have not considered the high reliability of the system from the perspective of actual operation. Under the architecture in which one CE consolidates all of connected FEs, if faults occur in a CE and the CE become unable to serve a routing function, all of the FEs controlled by the CE will be affected. FEs can continue to serve the data packet transmission during the fault of their CE because the FEs are physically-separated from the CE in this architecture. However, the FEs cannot receive the path computation and path setting from their CE. That is, FEs can no longer respond to changing situations of the network topology such as link-up/down of any ports, links, and FEs. Therefore, it is absolutely critical to ensure the high reliability of the CE in this architecture.

In this paper, we propose a fault recovery method in functionally distributed transport networking. This method responds to CE fault by taking the redundant configuration of N+m CEs. This means that network operators provide m standby CEs for the control-plane consisting of N active CEs. If faults occur in any of the active CEs, the operation of the failed CE will be continued in any of the standby CEs.

The paper is structured as follows: Section 2 gives an overview of functionally distributed transport networking. Section 3 proposes a fault recovery method of CEs. Section 4 explains how to implement a functionally distributed transport networking system and apply the proposed fault recovery method. Section 5 analyzes the fault recovery performance by using the implemented system in a simulation environment and evaluates its feasibility. Section 6 concludes the paper.

2 Overview of Functionally Distributed Transport Networking

2.1 Routing Framework

This section explains the basic framework of the functionally distributed transport networking architecture. Fig. 1 shows the configuration of an FDR in which a CE is physically-separated from an FE. In this model, one or more FEs connect to one CE. The CE controls all connected FEs via control sessions. The control sessions between the CE and the FEs are established through the management channels. There is no limit on the number of hops of the control session.

The router functions are distributed among the CE and the FEs.

1. The CE performs functions that:

- collect and manage the topology information of the network between the FEs and construct the LSDB.
- exchange the LSDB with other CEs that control other FE groups via the higher CE.
- compute the routing information base (RIB).
- generate the forwarding information base (FIB) of each connected FE from the RIB.
- send each FIB to the corresponding FE via the control session.

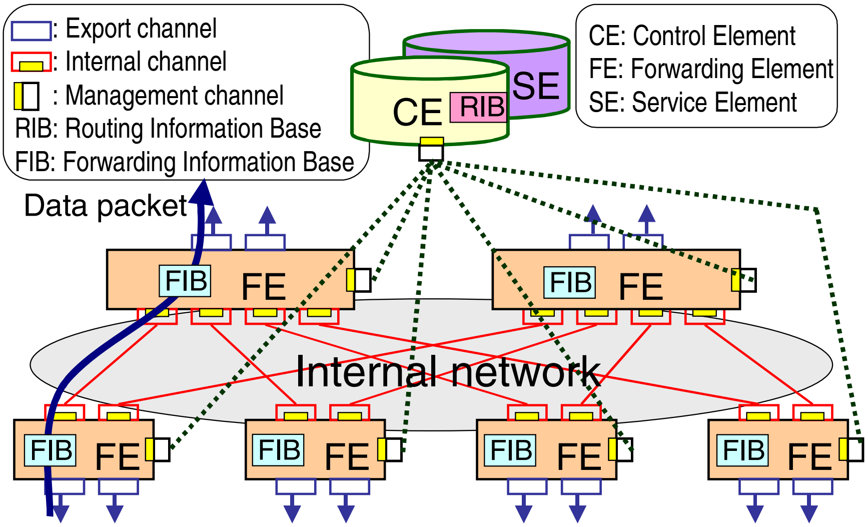


Fig. 1 Configuration of functionally distributed router

2. The FE performs functions that:

- collect topology information about the network between adjacent FEs.
- send topology information to the CE.
- forward the data packets based on the FIB.

2.2 *Expansion of Performance and Functions in Transport Stratum*

In functionally distributed transport networking architecture, the control-plane and the forwarding-plane are physically-separated, and the interface between those planes is specified. Therefore, performance and functions of each plane can be developed independently.

This architecture can also include a service element (SE) (see Fig. 1). This is an element in which the service processing (for example, security, monitoring, and access control) is executed on demand from the CE. The SE may be mounted on a control-plane processing server running any of CEs.

The number and capacities of CEs and SEs in the network can be increased according to the demand for processing performance. In addition, CEs can dynamically change the correspondence with the FEs in accordance with the load balancing policy.

3 Fault Recovery of CEs

It is important to ensure the high reliability of the CE in the architecture in which one CE consolidates and processes the path computations and the path settings for multiple FEs. Generally, current networks provide standby routers to respond to faults of active routers. In many cases, Virtual Router Redundancy Protocol (VRRP) [5] is adopted into the routers as a standardized protocol to switch from a failed router to a standby router. The standby router may be active for other VRRP group with a different VRID. Although VRRP switches automatically once a fault is detected on an active router, an active router and its standby routers have to be connected at the same LAN. This means that there is a restriction in which an active router connects to standby routers via one hop.

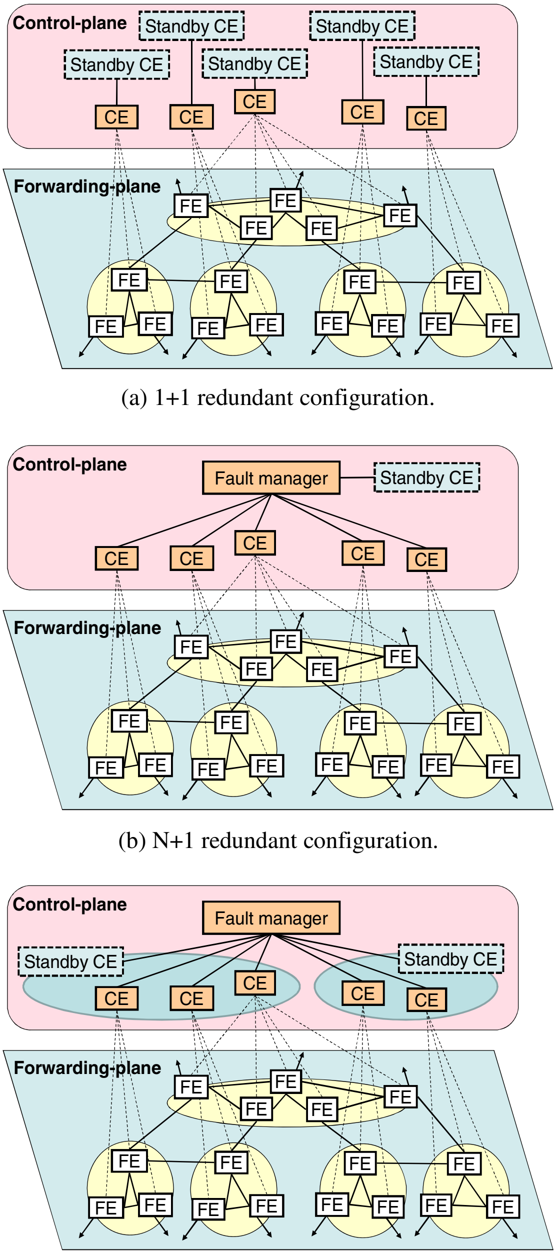
To respond to CE failover in functionally distributed transport networking architecture, we propose a CE failover method under the CE redundant configuration like VRRP. However, from the aspect of widely distributed elements in this architecture, the restriction on the number of hops between an active CE and a standby CE must be lifted.

3.1 CE Redundant Configuration

When network operators configure the control-plane with redundant CEs in their networks, they have to decide how many redundant CEs they need to provide and how to make a correlation between active CEs and standby CEs. There are several possible solutions depending on the requirements from the perspective of system reliability, configuration flexibility, and economical efficiency.

3.1.1 1+1 Redundant Configuration

1+1 redundant configuration is the simplest solution whereby each active CE has its own standby CE to ensure that CE functionality continues in the event of CE fault (see Fig. 2-(a)). Each active CE sends heartbeat signals to its own standby CE. When the standby CE no longer receives the heartbeat signals from the active CE, the standby CE judges the active CE to be in fault and takes over the operation as a new active CE. This configuration can achieve very high reliability. Because each subset of an active CE and a standby CE can operate failover process independently, the overall system integrity will not be impacted even when multiple active CEs break down at the same time. However, network operators have to prepare twice the number of CEs needed for primary operation. This means that this configuration is not economically efficient.



(a) 1+1 redundant configuration.

(b) N+1 redundant configuration.

(c) N+m redundant configuration.

Fig. 2 Redundant configurations

3.1.2 N+1 Redundant Configuration

N+1 redundant configuration is a solution in which multiple active CEs (N) have one independent standby CE to ensure that CE functionality continues in the event of CE fault (see Fig. 2-(b)). In this configuration, the system has a fault manager in the control-plane. The fault manager is connected with all of the active CEs and the standby CE. Each active CE sends heartbeat signals to the fault manager. When the fault manager no longer receives the heartbeat signals from any of the active CEs, the fault manager judges that active CE to be in fault and activates the standby CE as a new active CE. When the failed CE recovers, it connects to the fault manager as a new standby CE. Because only one standby CE is shared by all of the active CEs, network operators can keep extra expenses down. On the other hand, system reliability lowers. There is no standby CE during the recovery period of a failed CE. That is, the overall system integrity will be impacted when multiple active CEs break down at the same time.

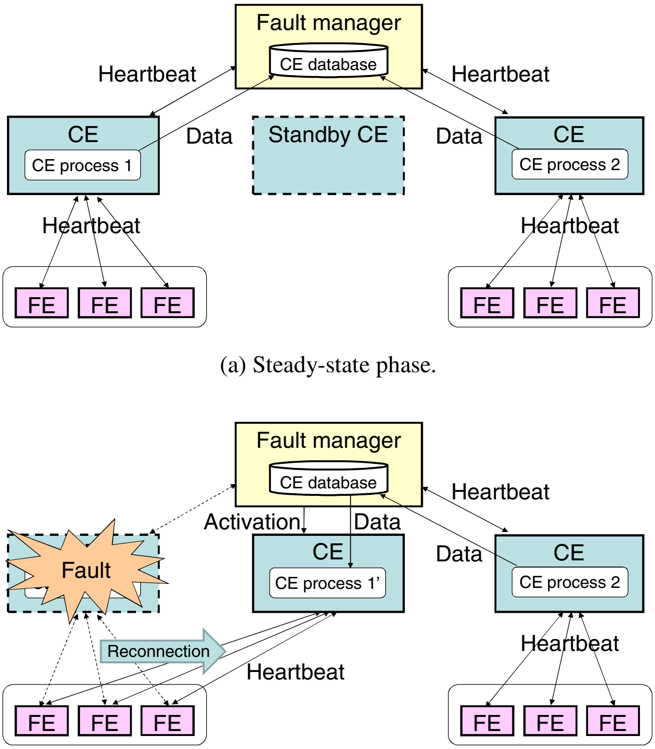
3.1.3 N+M Redundant Configuration

N+m redundant configuration is a solution in which multiple active CEs (N) have multiple independent standby CEs (m) to ensure that CE functionality continues in the event of CE fault (see Fig. 2-(c)). All active CEs are divided into several groups (m), and each CE group is related to its own standby CE. In this configuration, the system has a fault manager in the control-plane. The fault manager is connected with all of the active CEs and the standby CEs. Each active CE sends heartbeat signals to the fault manager. When the fault manager no longer receives the heartbeat signals from any of the active CEs, the fault manager judges that active CE to be in fault and activates the standby CE related to the group with the failed CE as a new active CE. When the failed CE recovers, it connects to the fault manager as a new standby CE for its group. This configuration results in a highly flexible redundant configuration. Network operators are free to decide the number of standby CEs in the control-plane depending on such requirements as CE fault rate and budget for deployment. If m is set to N, the 1+1 redundant configuration can be realized. If m is set to 1, the N+1 redundant configuration can be realized. Because of its flexibility, we propose to adopt this configuration into our CE failover method.

In this case, since there is a single fault manager that deals with all CEs, the fault manager is itself a single point of fault. This method requires full duplication of the fault manager for the robust system. A fault manager has an active body and a standby body and always runs status mirroring and data mirroring between two bodies. Also, links between a fault manager and CEs are duplicated. Because there is only one fault manager in whole system, its full duplication has little influence on economical efficiency. This paper supposes that a fault manager is guaranteed its correct operation, and focuses on CE faults.

3.2 Failover Operation

This section describes detailed failover operation under the N+m redundant configuration. Failover operation is attained with a combination of steady-state (see Fig. 3-(a)) and fault-state phases (see Fig. 3-(b)).



(a) Steady-state phase.

(b) Fault-state phase.

Fig. 3 Failover operation

In each phase, each component of the network, such as CE, FE and fault manager, plays an individual role. Unlike the popular distributed system such as the blade server which has all elements in one chassis, CEs are separated from FEs via a network in the proposed architecture. Therefore, separately from monitoring and switching CEs by a fault manager, FEs have to monitor the connected CE and reconnect to the new active CE. Since FEs move independently from a fault manager, the FE operation and the fault manager operation have to be integrated into one failover operation.

3.2.1 Steady-State Phase

The steady-state phase is a phase in which the system runs normally for routing and forwarding. In this phase, an active CE sends heartbeat signals to the fault manager and all of the FEs connected to that CE and uploads CE's control data including a list of connected FEs, LSDB and RIB to the CE database on the fault manager. The data is uploaded repeatedly in conjunction with data updates by the CEs to keep the CE database up-to-date. The fault manager monitors the health of the CEs by receiving heartbeat signals and stores the data from the CEs in the CE database. The fault manager also stores a table, which shows the correspondences between a group of several active CEs and a standby CE. The table is set to the fault manager by network operators. An FE monitors the health of the connected CE by receiving heartbeat signals as well as the fault manager. All the FEs are set to an address of the standby CE, to which they should reconnect in case they detect CE fault, by network operators.

3.2.2 Fault-State Phase

The fault-state phase is a phase in which the system recovers its incompleteness due to the CE fault by switching from the failed CE to the corresponding standby CE. When faults occur in any of the active CEs, the system goes into this phase. Due to breakdown of the heartbeat signals from the failed CE, the fault manager and FEs connected to that CE detect the CE fault. Then, the fault manager activates the standby CE related to the group with that failed CE as a new active CE and sets the data of the failed CE stored in the CE database to the new active CE. This enables the new active CE to quickly take over the operation. All of the FEs connected to the failed CE try to reconnect to the new active CE. As it may take some time before the new active CE completes preparations for the reconnection, FEs try to reconnect periodically. When the reconnections from all of the FEs are over, the new active CE collects the topology information of the network between the reconnected FEs and recomputes the LSDB, RIB and FIB of each reconnected FE. This enables the system to respond to changing situations in the network topology during a recovery period of the failed CE.

This operation assumes a basic "fail-stop" model for CE fault. In case that only the link between a fault manager and a CE has failed, the communication can recover quickly by switching to a standby link because links between a fault manager and CEs are duplicated. The same handling can be used for the case that only the link between a CE and an FE has failed.

4 System Implementation

To analyze the fault recovery performance based on the proposed method, we implemented a functionally distributed transport networking system and applied the proposed fault recovery method to the system. In this section, we explain our ways of implementing the system and applying the proposed method to the system.

4.1 Functionally Distributed Transport Networking System Implementation

4.1.1 XORP Retrofit

We implemented a functionally distributed transport networking system based on XORP [6, 7], which is a modular, extensible, open source IP routing software. XORP provides a fully featured platform that implements IPv4 and IPv6 routing protocols and a unified platform to configure them. Due to its modularity, we can easily add new functions. We retrofitted XORPv1.3 as follows.

1. To allow physical separation between a CE and an FE, we divided XORP functional modules into CE side and FE side, and packaged each of the module groups as a CE software and an FE software.
2. We used the ForCES [8, 9] protocol, which is specified by the Internet Engineering Task Force (IETF), as the communication interface between management channels of a CE and FEs. ForCES is a protocol by which CEs can configure and manage one or more separate FEs within a network element. There is no limit on the number of hops of the ForCES session between a CE and an FE. To apply the ForCES protocol, we added a ForCES communication management function to the CE and FE software.
3. Original XORP can control only one forwarding engine. We modified the CE software to control multiple FEs at a time. This enabled a CE to establish and maintain multiple ForCES sessions with each of FEs, manage and configure status of the connected FEs, and compute RIBs and FIBs of the connected FEs.
4. We added a higher CE mode to the CE software. The higher CE mode is used on a higher CE which treats each FDR composed of one CE and multiple FEs as one FE and computes optimal paths among the FDRs. A higher CE communicates with all of the CEs using XORP's extensible IPC mechanism called XORP Resource Locators (XRL).
5. We made a path computation module using the Shortest Path First (SPF) algorithm. This module can compute RIBs of all FEs based on an LSDB and generate FIBs of all FEs from the RIBs. Both CE and high CE recursively use the same path computation module to reduce the complexity of the path control.

4.1.2 Functional Architecture

Fig. 4 illustrates the functional architecture of our implementation. Higher CE, CE and FE have some function modules as follows.

1. The Higher CE has:

- CE management module which manages the information of the connected CEs, such as a CEID and an IP address.
- path computation module which receives the path information among all of the FDRs, maintains the LSDB, computes RIBs of all of the FDRs, and distributes the computed optimal path results to the corresponding FDRs.

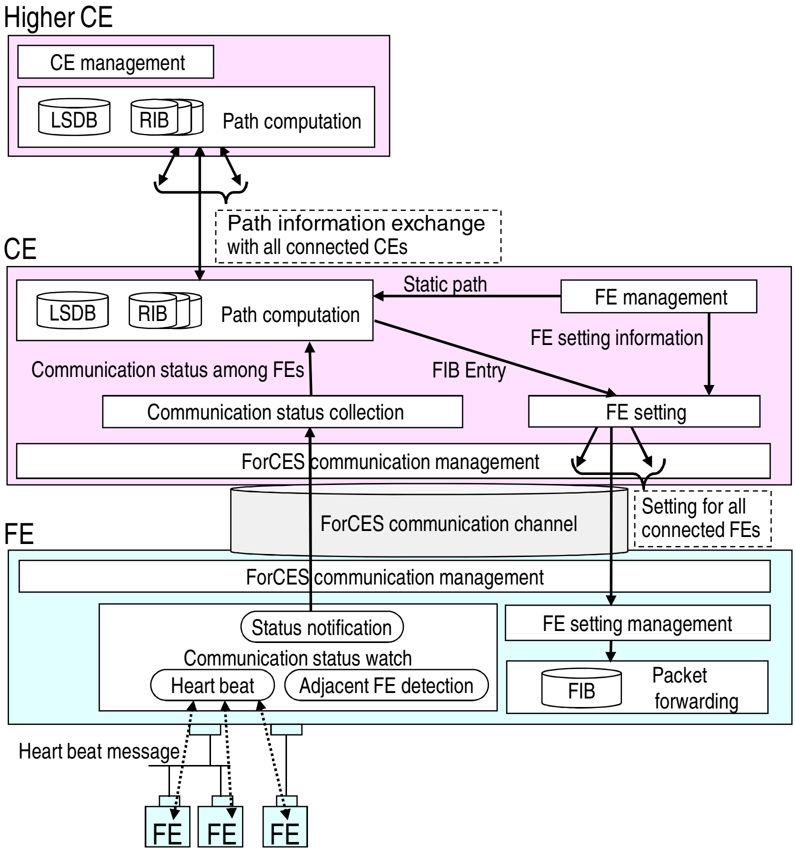


Fig. 4 Functional architecture of the implementation

- LSDB which is composed of the link state information among all of the FDRs, such as a network address and a metric.
- RIB of each FDR which is computed from the LSDB using the SPF algorithm.

2. The CE has:

- FE management module which manages the information of the connected FEs, such as an FEID and an IP address, passes the FE setting information to the FE setting module, and if any, passes the static path information among the FEs to the path computation module.
- FE setting module which receives the FE setting information from the FE management module and the FIB entries from the path computation module, and passes the information to the FE setting management module on the corresponding FEs via ForCES communication channels.
- path computation module which receives the path information among all of the connected FEs from the FE management module and the communication status

collection module, maintains the LSDB, computes RIBs of all of the connected FEs, generates FIBs of all of the connected FEs, and passes the generated FIB entries to the FE setting module.

- LSDB which is composed of the link state information among all of the connected FEs, such as a network address and a metric.
- RIB of each connected FE which is computed from the LSDB using the SPF algorithm.
- communication status collection module which receives the communication status among FEs including the link state information from the communication status watch module on each FE via ForCES communication channel, and passes the information to the path computation module.
- ForCES communication management module which establishes and maintains the ForCES communication channels between a CE and each connected FE, encapsulates a message between a CE and each connected FE with the ForCES massage format, and exchange the encapsulated massage via the ForCES communication channel.

3. The FE has:

- FE setting management module which receives the FE setting information and the FIB entries from the FE setting module on the connected CE, reflects the FE setting information to the FE operation, and passes the FIB entries to the packet forwarding module.
- packet forwarding module which receives the FIB entries from the FE setting management module, stores the FIB entries on the FIB, and forwards the data packets based on the FIB.
- FIB which is composed of the FIB entries set by the FE setting module on the connected CE.
- communication status watch module which detects adjacent FEs, monitors the communication status with the adjacent FEs by exchanging heartbeat massages, and notifies the communication status collection module on the connected CE of the latest communication status.
- ForCES communication management module which is same as that of a CE.

4.2 Application of Fault Recovery Method

We applied the proposed fault recover method to the implemented system described above. To allow the implemented system to operate as shown in fig.3, we added the functions for the failover operation as follows.

1. We added the fault manager module to the higher CE. The fault manager exchanges heartbeat signals with all of the active CE, maintains the CE database, manages the correspondences between a group of active CEs and a standby CE, activates the appropriate standby CE when faults occur in any of the active CEs, and sets the data of the failed CE stored in the CE database to the new active CE.

2. We added the CE high availability module to the CE. The CE high availability module exchanges heartbeat signals with the higher CE and all of the connected FEs, and sends CE's control data to the fault manager module on the high CE.
3. We added the FE high availability module to the FE. The FE high availability module exchanges heartbeat signals with the connected CE, stores an address of the standby CE set by network operators, and reconnects to the new active CE when faults occur in the connected CE.

5 Fault Recovery Performance Analysis

Under the architecture in which one CE consolidates all connected FEs, if faults occur in a CE and it becomes unable to serve a routing function, all of the FEs controlled by that CE cannot receive the path computation and path setting from their CE. That is, FEs can no longer respond to changing situations in the network topology such as link-up/down of any ports, links, and FEs. Therefore, CE failover operation must be completed as quickly as possible.

To evaluate the feasibility of the proposed fault recovery method, we determined the time of CE failover operation by using our implementation in a simulation environment. In the simulation environment, we used two servers with two dual-core Xeon (2.13 GHz) processors for an active and a standby CEs. We prepared two more servers for a higher CE and FEs. The FE server can run multiple FE processes at the same time, and each FE process can establish an individual association to the CE process. All of the servers in the simulation environment run on the CentOS 5 [10] operating system, which is a Linux distribution derived from sources provided free to the public. The bandwidth of all links between the servers is 1.0 Gbps. Since the bandwidth is sufficiently-broad to exchange the massages between servers, there is no bottleneck link.

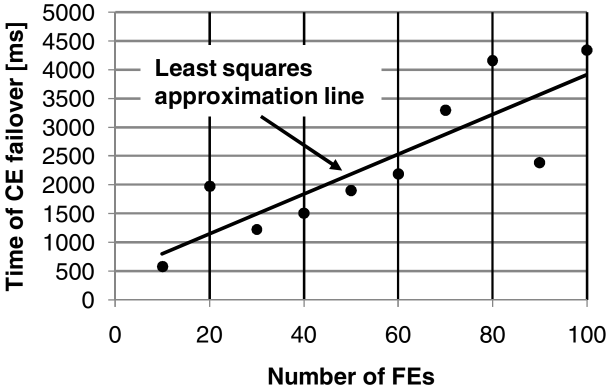


Fig. 5 CE-failover time

The simulation is supposed to start when the fault manager module on the higher CE detects the fault of the active CE. This detection induces the activation of the standby CE and setting the data of the failed CE stored in the CE database to the new active CE. Then, all of the FEs reconnect to the new active CE. The new active CE processes the ForCES association setup for all of the FEs. As for time of CE failover, we determined the time from CE fault detection by the fault manager module to the completion of the reconnection by all the FEs. Additionally, the simulation was carried out on various network scales. There were ten network topologies wherein the number of FEs changed from 10 to 100 in steps of 10. As each FE has 10 links, the number of links in whole system changed from 100 to 1000 in steps of 100.

Fig. 5 shows the results from the CE failover. This scatter diagram plots CE failover on various network scales, and an approximation line was made using the least squares method. Because the new active CE has to process the requests of the reconnection equaling the number of FEs, the total time increases dramatically with the number of FEs. Additionally, an increase in the number of FEs means increased possibility of FE reconnection fault and retry. Several reconnection requests at once ran out of time. This leads to an increase in the total processing time. However, the graph is not in perfect incremental shape. This is attributed to the timers set in the algorithm. We set the interval parameters of the CE fault detection and the reconnection retry by an FE to 1.0 s. It would appear that those timers caused the variation in CE-failover time. Even if the number of FEs was 100, CE-failover time would be less than 5 s. Given that the initial value of RouterDeadInterval in OSPFv2 [11] is 40 s, the CE-failover time is short enough not to be recognized as a fault by neighbor routers. Therefore, we can say that the proposed method has adequate feasibility.

For a further study, an experimental evaluation of the feasibility and scalability of the proposed method in real environments is important issue.

6 Conclusion

We proposed a fault recovery method in functionally distributed transport networking, which leads to reduction in the path-control complexity and efficient operation of large scale networks. Under the architecture in which one CE consolidates all connected FEs, if faults occur in a CE and the CE become unable to serve a routing function, all the FEs controlled by that CE cannot receive the path computation and path setting from their CE. Therefore, it is absolutely critical to ensure the high reliability of the CE in this architecture. Our proposed method responds to CE faults by taking the redundant configuration of N+m CEs. This means that network operators provide m standby CEs for the control-plane consisting of N active CEs. If faults occur in any of the active CEs, the failed CE function will be continued in any of the standby CEs. Additionally, we described the operation of each component in the proposed method, explained how to implement a functionally distributed transport networking system and apply the proposed fault recovery method, and analyzed the fault recovery performance by using the implemented system in a simulation environment. We confirmed that the time of CE failover is short enough and the proposed fault recovery method has adequate feasibility.

Acknowledgments. This work was supported in part by the National Institute of Information and Communications Technology (NICT) of Japan.

References

1. Lakshman, T.V., Nandagopal, T., Ramjee, R., Sabnani, K., Woo, T.: The Sof-trouter Architecture. In: ACM HotNets-III Workshop (2004)
2. Aoki, M., Ogawa, K., Hamano, T., Chaki, S.: Functionally Distributed Trans-port Networking on Next-Generation Network. In: IEEE-WS NGN-EC2, pp. 593–600 (2007)
3. Ogawa, K., Aoki, M., Chaki, S.: Verification of Path Computational Perform-ance in Functionally Distributed Transport Networking on Next-Generation Network. In: IEICE APSITT 2008, pp. A-1-2 (2008)
4. OpenFlow, `http://www.openflowswitch.org/wp/documents/`
5. Hinden, R.: Virtual Router Redundancy Protocol (VRRP). Internet Engineering Task Force, RFC3768, `http://ftp.ietf.org/rfc/rfc3768.txt`
6. Handley, M., Hodson, O., Kohler, E.: XORP: An Open Platform for Network Research. ACM SIGCOMM, Computer Communication Review 33(1), 53–57 (2003)
7. XORP Project, `http://www.xorp.org`
8. Khosravi, H., Anderson, T.: Requirements for Separation of IP Control and Forwarding. Internet Engineering Task Force, RFC3654, `ftp://ftp.ietf.org/rfc/rfc3654.txt`
9. Yang, L., Dantu, R., Anderson, T.: Forwarding and Control Element Separation (ForCES) Framework. Internet Engineering Task Force, RFC3764, `ftp://ftp.ietf.org/rfc/rfc3746.txt`
10. The Community ENTerprise Operating System, `http://centos.org`
11. Moy, J.: OSPF Version 2. Internet Engineering Task Force, RFC2328, `ftp://ftp.ietf.org/rfc/rfc2328.txt`

Author Index