

ADVANCED METHODS AND TECHNOLOGIES FOR AGENT AND MULTI-AGENT SYSTEMS

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including “Information Modelling and Knowledge Bases” and “Knowledge-Based Intelligent Engineering Systems”. It also includes the biennial ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Series Editors:

J. Breuker, N. Guarino, J.N. Kok, J. Liu, R. López de Mántaras,
R. Mizoguchi, M. Musen, S.K. Pal and N. Zhong

Volume 252

Recently published in this series

- Vol. 251. P. Vojtás, Y. Kiyoki, H. Jaakkola, T. Tokuda and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXIV
- Vol. 250. B. Schäfer (Ed.), Legal Knowledge and Information Systems – JURIX 2012: The Twenty-Fifth Annual Conference
- Vol. 249. A. Caplinskas, G. Dzemyda, A. Lipeikiene and O. Vasilecas (Eds.), Databases and Information Systems VII – Selected Papers from the Tenth International Baltic Conference, DB&IS 2012
- Vol. 248. D. Riaño, E. Onaindia and M. Cazorla (Eds.), Artificial Intelligence Research and Development – Proceedings of the 15th International Conference of the Catalan Association for Artificial Intelligence
- Vol. 247. A. Tavast, K. Muischnek and M. Koit (Eds.), Human Language Technologies – The Baltic Perspective – Proceedings of the Fifth International Conference Baltic HLT 2012
- Vol. 246. H. Fujita and R. Revetria (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the Eleventh SoMeT_12
- Vol. 245. B. Verheij, S. Szeider and S. Woltran (Eds.), Computational Models of Argument – Proceedings of COMMA 2012
- Vol. 244. S. Scheider, Grounding Geographic Information in Perceptual Operations
- Vol. 243. M. Graña, C. Toro, J. Posada, R.J. Howlett and L.C. Jain (Eds.), Advances in Knowledge-Based and Intelligent Information and Engineering Systems
- Vol. 242. L. De Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz and P. Lucas (Eds.), ECAI 2012 – 20th European Conference on Artificial Intelligence
- Vol. 241. K. Kersting and M. Toussaint (Eds.), STAIRS 2012 – Proceedings of the Sixth Starting AI Researchers’ Symposium

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Advanced Methods and Technologies for Agent and Multi-Agent Systems

Edited by

Dariusz Barbucha

Department of Information Systems, Gdynia Maritime University, Poland

Manh Thanh Le

Hue University, Vietnam

Robert J. Howlett

Bournemouth University, United Kingdom

and

Lakhmi C. Jain

University of South Australia, Australia

IOS
Press

Amsterdam • Berlin • Tokyo • Washington, DC

© 2013 The authors and IOS Press.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 978-1-61499-253-0 (print)

ISBN 978-1-61499-254-7 (online)

Library of Congress Control Number: 2013938422

Publisher

IOS Press BV

Nieuwe Hemweg 6B

1013 BG Amsterdam

Netherlands

fax: +31 20 687 0019

e-mail: order@iospress.nl

Distributor in the USA and Canada

IOS Press, Inc.

4502 Rachael Manor Drive

Fairfax, VA 22032

USA

fax: +1 703 323 3668

e-mail: iosbooks@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

This volume contains the proceedings of the 7th KES Conference on Agent and Multi-Agent Systems – Technologies and Applications (KES-AMSTA 2013) held on May 27–29, 2013 in Hue city, Vietnam. The conference was organized by KES International and its Focus Group on Agent and Multi-agent Systems, and Hue University, Vietnam.

Following the success of previous KES Symposia/Conferences on Agent and Multi-Agent Systems – Technologies and Applications, held in Wroclaw, Poland (KES-AMSTA 2007), Incheon, Korea (KES-AMSTA 2008), Uppsala, Sweden (KES-AMSTA 2009), Gdynia, Poland (KES-AMSTA 2010), Manchester, UK (KES-AMSTA 2011), and Dubrovnik, Croatia (KES-AMSTA 2012), this conference continues to provide an internationally respected forum for scientific research in the technologies and applications of agent and multi-agent systems.

The field of agent and multi-agent systems is concerned with the development and evaluation of sophisticated, AI-based problem-solving and control architectures for both single-agent and multi-agent systems. Current topics of research in the field include, among other, agent-oriented software engineering, beliefs engineering, desires and intentions representation, agent co-operation, co-ordination, negotiation, organization and communication, distributed problem solving, specification of agent communication languages, formalization of ontologies and conversational agents. Special attention is paid to the feature topics: Intelligent technologies and applications in the area of e-health, social networking, self-organizing systems, economics and trust management.

KES-AMSTA 2013 features a number of keynote talks, oral presentations, and invited sessions, closely aligned to the theme of the conference. The conference attracted a substantial number of researchers and practitioners from all over the world who submitted their papers for the main track covering the methodology and applications of agent and multi-agent systems, two special sessions on specific topics within the field, and a half day workshop for early career stage researchers.

The main track streams, covering the methodology and applications of agents and multi-agent systems, includes sessions on: Multi-Agent Systems Design and Implementation, Agent-Based Modeling and Simulation, Coordination, Cooperation and Teamwork, Agent-Based Optimization, Web Services and Semantic Web, Agent Theories, Models and Communication, and Social and Business Issues.

In addition to the main tracks of the conference two invited sessions are hosted: Intelligent Agents with Semantic Technology (IAST 2013), and Computational Intelligence for Business Collaboration (CIBC 2013). On the final day of the conference the workshop entitled New Directions in Agents Research is held.

Submissions to KES-AMSTA 2013 came from 20 countries. Each paper was peer reviewed by at least two members of the International Programme Committee and International Reviewer Board. In all, 44 best papers were selected for oral presentation and publication in the proceedings volume of KES-AMSTA 2013.

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

We would like to express our sincere thanks to the KES-AMSTA 2013 General Chair and KES-AMSTA Symposium Series and Focus Group on Agent and Multi-agent Systems Chair, Ngoc Thanh Nguyen from Wroclaw University of Technology, Poland. Since the first KES-AMSTA conference in 2007, each year his work and involvement in the organization of KES-AMSTA series conferences has contributed to ensure high quality of these events.

We are very grateful to the keynote speakers, Andrzej Szalas, University of Warsaw, Poland, and Arkady Zaslavsky, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia, for their interesting and informative talks of the world-class standard.

Our thanks are due to all International Programme Committee members for their valuable efforts in the review process, which helped us to guarantee the highest quality of selected papers for the conference. Thanks are also due to several additional reviewers, who contributed to the conference success. Our special gratitude is also directed to the Invited Session Chair, Linh Anh Nguyen, and organizers and chairs of invited sessions, Trong Hai Duong, Jason J. Jung, and Hanh H. Hoang, for their valuable contribution.

We extend our thanks to main organizers and sponsors, KES International and Hue University, Vietnam. Our special thanks go to the Organizing Chair, Hanh H. Hoang, and all members of Local Organizing Committee for their excellent efforts in the organizational work.

Finally, we cordially thank all the authors for their valuable contributions and the other participants in this conference. The conference would not have been possible without their support.

March 2013

Dariusz Barbucha
Manh Thanh Le
Robert J. Howlett
Lakhmi C. Jain

International Programme Committee

- Ahmad Taher Azar, IGI Global, USA
- Messaouda Azzouzi, Faculty of Sciences and Technology, Ziane Achour University of Djelfa, Algeria
- Costin Badica, University of Craiova, Romania
- Dariusz Barbucha, Gdynia Maritime University, Poland
- Andrej Brodnik, University of Ljubljana and University of Primorska, Slovenia
- Grażyna Brzykcy, Poznan University of Technology, Poland
- Longbing Cao, University of Technology Sydney, Australia
- Frantisek Capkovic, Slovak Academy of Sciences, Slovak Republic
- Krzysztof Cetnarowicz, AGH University of Science and Technology, Poland
- Ireneusz Czarnowski, Gdynia Maritime University, Poland
- Paul Davidsson, Malmo University, Sweden
- Barbara Dunin-Keplicz, Warsaw University and Polish Academy of Sciences, Poland
- Trong Hai Duong, Quangbinh University, Vietnam
- Arnulfo-Alanis Garza, Instituto Tecnológico de Tijuana, México
- Lilia Georgieva, Heriot-Watt University, UK
- Paulina Golinska, Poznan University of Technology, Poland
- Anne Hakansson, KTH Royal Institute of Technology, Sweden
- Chihab Hanachi, University of Toulouse 1 Capitole - IRIT Laboratory , France
- Ronald L. Hartung, Franklin University, USA
- Huu-Hanh Hoang, Hue University, Vietnam
- Mirjana Ivanovic, University of Novi Sad, Serbia
- Adam Jatowt, Kyoto University, Japan
- Joanna Jedrzejowicz, University of Gdansk, Poland
- Piotr Jedrzejowicz, Gdynia Maritime University, Poland
- Dragan Jevtic, University of Zagreb, Croatia
- Gordan Jezic, Unviersity of Zagreb, Croatia
- Jason J. Jung, Yeungnam University, Korea
- Radoslaw Piotr Katarzyniak, Wroclaw University of Technology, Poland
- Arkadiusz Kawa, Poznam University of Economics, Poland
- Adrianna Kozierkiewicz-Hetmanska, Wroclaw University of Technology, Poland
- Dariusz Krol, Wroclaw University of Technology, Poland
- Konrad Kułakowski, AGH University of Science and Technology, Poland
- Mario Kusek, University of Zagreb, Croatia
- Kazuhiro Kuwabara, Ritsumeikan University, Japan
- Fuhua Lin, Athabasca University, Canada
- Marin Lujak, Universidad Rey Juan Carlos, Spain
- Manuel Mazzara, Newcastle University, UK
- Daniel Moldt, University of Hamburg, Germany
- Ngoc Thanh Nguyen, Wroclaw University of Technology, Poland

- James O'Shea, Manchester Metropolitan University, UK
- Vedran Podobnik, University of Zagreb, Croatia
- Bhanu Prasad, Florida A&M University, USA
- Radu-Emil Precup, Politehnica University of Timisoara, Romania
- Ewa Ratajczak-Ropel, Gdynia Maritime University, Poland
- Adam Sędziwy, AGH University of Science and Technology, Poland
- Alexei Sharpanskykh, VU University Amsterdam, Netherlands
- Andrzej Szalas, University of Warsaw, Poland
- Wojciech Thomas, Wroclaw University of Technology, Poland
- Bogdan Trawinski, Wroclaw University of Technology, Poland
- Krunoslav Trzec, Ericsson Nikola Tesla, Croatia
- Bay Vo, Information Technology College, Vietnam
- Toyohide Watanabe, Nagoya University, Japan
- Izabela Wierzbowska, Gdynia Maritime University, Poland
- Mahdi Zargayouna, IFSTTAR, France
- Arkady Zaslavsky, CSIRO ICT Centre, Australia

International Reviewer Board

- Angelo Di Iorio
- Jerome Euzenat
- Emanuela Goldoni
- Nojeong Heo
- Sanggil Kang
- Sang Boem Lim
- Sangyoон Oh
- Nafees Qamar
- Cinzia Spina
- Jieh-Shan Yeh
- Hai Zhuge

Contents

Preface	v
---------	---

Keynote

Partiality and Inconsistency in Agents' Belief Bases <i>Jan Małuszyński and Andrzej Szałas</i>	3
---	---

Multi-Agent Systems Design and Implementation

Erlang as a High Performance Software Agent Platform <i>Wojciech Turek</i>	21
---	----

Travel Routes Flexibility in Transport Systems <i>František Čapkovíč</i>	30
---	----

On Scalable, Event-Oriented Control for Lighting Systems <i>Igor Wojnicki, Leszek Kotulski and Sebastian Ernst</i>	40
---	----

Emergence of Collective Escaping Strategies in Caribou Agents <i>Kun Tian, Ivan Tanev and Katsunori Shimohara</i>	50
--	----

On Acceleration of Multi-Agent System Performance in Large Scale Photometric Computations <i>Adam Sędzwiw</i>	58
--	----

Agent-Based Modeling and Simulation

Modeling Value Co-Creation Process in Complex Service Systems Using Kauffman's NKCS Architecture <i>Chathura Rajapakse and Takao Terano</i>	71
--	----

Agent-Based Simulator for Travelers Multimodal Mobility <i>Mahdi Zargayouna, Besma Zeddini, Gérard Scemama and Amine Othman</i>	81
--	----

Multi-Level Agent-Based Modeling: A Generic Approach and an Implementation <i>Dirc-Ân Võ, Alexis Drogoul and Jean-Daniel Zucker</i>	91
--	----

Optimizing an Environmental Surveillance Network with Gaussian Process Entropy – An Optimization Approach by Agent-Based Simulation <i>Viet Xuan Truong, Hiep Xuan Huynh, Minh Ngoc Le and Alexis Drogoul</i>	102
--	-----

Coordination, Cooperation and Teamwork

- Obligation and Prohibition Norms Mining Algorithm for Normative Multi-Agent Systems 115

*Moamin A. Mahmoud, Mohd Sharifuddin Ahmad, Azhana Ahmad,
Mohd Zaliman M. Yusoff, Aida Mustapha and
Nurzeatul Hamimah Abdul Hamid*

- Role and Member Selection in Team Formation Using Resource Estimation 125
Masashi Hayano, Dai Hamano and Toshiharu Sugawara

- Incorporating Explicit Coordination Mechanisms by Agents to Obtain Green Waves 137

Antonio de Abreu Batista Júnior and Luciano Reis Coutinho

- Communication Leading to Coalition Nash Equilibrium I 146
Takashi Matsuhisa

Agent-Based Optimization

- An Agent-Based Cooperative Population Learning Algorithm for Vehicle Routing Problem with Time Windows 159

Dariusz Barbucha

- Mobile Agent-Based Dynamic Resource Allocation Method for Query Optimization in Data Grid Systems 169

Igor Epimakhov, Abdelkader Hameurlain, Franck Morvan and Shaoyi Yin

- Memetic Multi-Agent Computing in Difficult Continuous Optimisation 181
Aleksander Byrski, Wojciech Korczyński and Marek Kisiel-Dorohinicki

Web Services and Semantic Web

- A Technical Survey for Linked Open Data Federation Building 193

Cuong Duc Nguyen and Trong Hai Duong

- A Deduction-Based System for Formal Verification of Agent-Ready Web Services 203

Radosław Klimek

- Software Bug Ontology Supporting Bug Search on Peer-to-Peer Networks 213
Ha Manh Tran, Son Thanh Le, Synh Viet Uyen Ha and Tu Kha Huynh

Agent Theories, Models and Communication

- On the Horn Fragments of Serial Regular Grammar Logics with Converse 225
Linh Anh Nguyen and Andrzej Szałas

Cut-Free ExpTime Tableaux for Converse-PDL Extended with Regular Inclusion Axioms <i>Linh Anh Nguyen</i>	235
Abstract Test Suite Specification for ACL Communicating Agents <i>Marina Bagić Babac and Dragan Jevtić</i>	245
Perceiving Speech Acts under Incomplete and Inconsistent Information <i>Barbara Dunin-Kęplicz, Alina Strachocka, Andrzej Szałas and Rineke Verbrugge</i>	255
Social and Business Issues	
The Agent-Based Diffusion Model on a Combined Social Network <i>Setsuya Kurahashi</i>	267
Towards the Validation of Agent-Based BPM Simulation <i>Roman Šperka, Dominik Vymětal and Marek Spišák</i>	276
The Supply Chain in Cloud Computing – the Natural Future <i>Katarzyna Grzybowska, Gábor Kovács and Balázs Lénárt</i>	284
Incentive Contracts in Logistics Outsourcing <i>Qin Zhu and Richard Y.K. Fung</i>	293
A Multi-Agent System for Games Trading on B2B Market Based on Users' Skills and Preferences <i>Pavle Skocir and Gordan Jezic</i>	303
Intelligent Agents with Semantic Technology (IAST 2013)	
Automatic Reuse of Interaction Protocols in Mas: Arip Model <i>Sami Khalfaoui and Wided Lejouad Chaari</i>	315
A Method for Knowledge Integration Using Indeterminate Model of Time with Criterion O ₂ <i>Van Du Nguyen and Ngoc Thanh Nguyen</i>	325
A Layered Adjustable Autonomy Approach for Dynamic Autonomy Distribution <i>Salama A. Mostafa, Mohd Sharifuddin Ahmad, Muthukkaruppan Annamalai, Azhana Ahmad and Ghussoon Salim Basheer</i>	335
K-depth RDF Keyword Search Algorithm Based on Structure Indexing <i>Minho Bae, Duc Nguyen, Sanggil Kang and Sangyoon Oh</i>	346
An Efficient Method for Hiding High Utility Itemsets <i>Bay Vo, Chun-Wei Lin, Tzung-Pei Hong, Vinh V. Vu, Minh Nguyen and Bac Le</i>	356

A Method for the Generation of Semantic Annotation from Sport News Using Ontology Based Patterns <i>Quang-Minh Nguyen, Tuan-Dung Cao, Thanh-Hien Phan, Hoang-Cong Nguyen and Tatsuya Hagino</i>	364
A Multi-Agent Architecture for Health Information Systems <i>Luca Palazzo, Matteo Rossi, Aldo Franco Dragoni, Andrea Claudi, Gianluca Dolcini and Paolo Sernani</i>	375
Web Service Composition with Petri Net and Ontology <i>Azizbek Marakhimov, Jaegeol Yim and Jaehun Joo</i>	385
A New BDI Architecture To Formalize Cognitive Agent Behaviors Into Simulations <i>Van-Minh Le, Benoit Gaudou, Patrick Taillandier and Duc-An Vo</i>	395
Agent-Based Military Engagement Simulation System Architecture with Implementation <i>Won K. Hwam, Yongho Chung, Junghoon Kim and Sang C. Park</i>	404
Computational Intelligence for Business Collaboration (CIBC 2013)	
Relationships Among the Concepts of Reduct in Incomplete Decision Tables <i>Nguyen Long Giang and Vu Van Dinh</i>	417
Exploiting Linked Open Data for Attribute Selection on Recommendation Systems <i>Xuan Hau Pham, Jason J. Jung and Hideaki Takeda</i>	427
Semantic Service Matchmaking for Ad Hoc Supply Chain Formation: A Network Analysis Approach <i>Duc Nguyen Trung and Jason J. Jung</i>	434
A Combination of Business Rule and Modeling Languages for Semantic Business Processes Modeling <i>Chi P.T. Tran and Hanh Huu Hoang</i>	444
An Ontological Approach for Collaborative Business Processes Formulation with Consensus Methodology <i>Trung Van Nguyen, Minh T.N. Hoang and Hanh Huu Hoang</i>	454
Subject Index	465
Author Index	469

Keynote

Partiality and Inconsistency in Agents' Belief Bases¹

Jan MAŁUSZYŃSKI^a and Andrzej SZAŁAS^{b,2}

^a Department of Computer and Information Science, Linköping University, Sweden

^b Institute of Informatics, University of Warsaw, Poland

and Department of Computer and Information Science, Linköping University, Sweden

Abstract. Agents' beliefs can be incomplete and partially inconsistent. The process of agents' belief formation in such contexts has to be supported by suitable tools allowing one to express a variety of inconsistency resolving and nonmonotonic reasoning techniques.

In this paper we discuss 4QL*, a general purpose rule-based query language allowing one to use rules with negation in the premises and in the conclusions of rules. It is based on a simple and intuitive semantics and provides uniform tools for lightweight versions of well-known forms of nonmonotonic reasoning. In addition, it is tractable w.r.t. data complexity and captures PTIME queries, so can be used in real-world applications.

Reasoning in 4QL* is based on well-supported models. We simplify and at the same time generalize previous definitions of well-supported models and develop a new algorithm for computing such models.

Keywords. belief bases, rule-based query languages, agent systems

1. Introduction and Motivations

In this paper we discuss agents' belief bases with partially settled and/or inconsistent beliefs. As the underlying tool we use 4QL*, an extension of 4QL query language being a DATALOG^{¬¬}-like language introduced in [1,2]. 4QL* allows one to use rules with negation in premises and conclusions of rules. It is based on a simple and intuitive semantics and provides uniform tools for lightweight versions of well-known forms of nonmonotonic/defeasible reasoning. In addition, 4QL* is tractable w.r.t. data complexity and captures PTIME queries. In the context of agents' belief bases its predecessor 4QL has been used in [3,4].

Agents' beliefs can be incomplete and partially inconsistent. As discussed in [3], the process of agents' belief formation begins with belief acquisition. Such initial beliefs are then transformed into final beliefs according to agents' epistemic profiles, using a variety of inconsistency resolving and nonmonotonic reasoning techniques. To illustrate the ideas consider the following scenario:

¹Supported by the Polish National Science Centre grant 2011/01/B/ST6/02769.

²Corresponding Author: Andrzej Szalas, Institute of Informatics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland; E-mail: andrzej.szalas@mimuw.edu.pl.

- A suspect is guilty if there is an evidence that he committed a crime.
A suspect is not guilty if he has an alibi. In the investigation phase a data base of evidences and alibis is created for each suspect.
At the trial a suspect is sentenced to jail if he is found guilty, he is freed if he is found not-guilty or it is unknown whether he is guilty, and the case is sent to further investigation if there is both a crime evidence and an alibi. (1)

In the scenario we have a rule with a negative conclusion: “a suspect is not guilty if he has an alibi”, a rule for dealing with lack of information: “a suspect is freed if it is unknown whether he is guilty” and a rule for resolving inconsistency: “further investigation if there is both a crime evidence and an alibi”. Designing such, and much more complex belief bases, requires suitable tools. In this paper we advocate for 4QL* as a proper tool: it allows one to design agents’ belief bases using rules, modules and remote access to relations via external literals. It still enjoys the important properties of 4QL: it is tractable w.r.t. data complexity and captures PTIME queries.

The major semantical concept of 4QL and 4QL* is the well-supported model. Queries to belief bases are evaluated in such models. Intuitively, a model for a set of rules is well-supported if all conclusions it contains are justified by applying rules, where the reasoning process is grounded in facts.

In the current paper we provide a simplified and at the same time generalized definition of well-supported models and develop a new algorithm for computing these models. The new algorithm is simpler and easier to understand, and corrects inconsistency propagation which has not been fully solved in the algorithm developed in [1,2]. From the algorithm it is clearly visible that inconsistencies, though do not violate tractability, are computationally more demanding than lack of information.³

The rest of the paper is structured as follows. In Section 2 we recall the logic we use. Next, in Section 3, we define the 4QL* language extending 4QL by allowing multiple rules with the same conclusion and discuss their status. Sections 4 and 5 are devoted to well-supported models and an algorithm to compute them. In Section 6 we discuss modules and external literals. Section 7 concludes the paper.

2. The Underlying Logic

We assume four truth values: f (false), u (unknown), i (inconsistent), t (true). These values naturally reflect scenarios, where many information sources may provide inconsistent information, being at the same time incapable of supplying whole necessary information. The following definition underlies modeling principles frequently used when multiple information sources are present.

Definition 2.1 By a *literal* we understand any expression of the form $r(\bar{t})$ or $\neg r(\bar{t})$, where r is a relation symbol and \bar{t} is a tuple consisting of constants and/or variables. A *ground literal* is a literal without variables.

By an *interpretation* we mean any finite set of ground literals. The *truth value* of a literal ℓ in interpretation \mathcal{I} , denoted by $\mathcal{I}(\ell)$, is the value defined as follows:

³This is an important methodological observation, allowing to design more efficient 4QL-based belief bases.

$$\mathcal{I}(\ell) \stackrel{\text{def}}{=} \begin{cases} \mathbf{t} & \text{if } \ell \in \mathcal{I} \text{ and } (\neg\ell) \notin \mathcal{I}; \\ \mathbf{i} & \text{if } \ell \in \mathcal{I} \text{ and } (\neg\ell) \in \mathcal{I}; \\ \mathbf{u} & \text{if } \ell \notin \mathcal{I} \text{ and } (\neg\ell) \notin \mathcal{I}; \\ \mathbf{f} & \text{if } \ell \notin \mathcal{I} \text{ and } (\neg\ell) \in \mathcal{I}. \end{cases}$$

△

We accept the following order on truth values (for the discussion and comparison with other approaches see [5,1,2,6,7]):

$$\mathbf{f} < \mathbf{u} < \mathbf{i} < \mathbf{t}. \quad (2)$$

Disjunction and conjunction are respectively maximum and minimum w.r.t. (2). The semantics of propositional connectives is given in Table 1.

Table 1. Truth tables for \wedge , \vee , \rightarrow and \neg .

\wedge	\mathbf{f}	\mathbf{u}	\mathbf{i}	\mathbf{t}	\vee	\mathbf{f}	\mathbf{u}	\mathbf{i}	\mathbf{t}	\rightarrow	\mathbf{f}	\mathbf{u}	\mathbf{i}	\mathbf{t}	\neg	
\mathbf{f}	\mathbf{i}	\mathbf{t}	\mathbf{f}	\mathbf{t}	\mathbf{t}	\mathbf{t}	\mathbf{t}	\mathbf{f}	\mathbf{t}							
\mathbf{u}	\mathbf{f}	\mathbf{u}	\mathbf{u}	\mathbf{u}	\mathbf{u}	\mathbf{u}	\mathbf{u}	\mathbf{i}	\mathbf{t}	\mathbf{u}	\mathbf{t}	\mathbf{t}	\mathbf{t}	\mathbf{t}	\mathbf{u}	\mathbf{u}
\mathbf{i}	\mathbf{f}	\mathbf{u}	\mathbf{i}	\mathbf{i}	\mathbf{i}	\mathbf{i}	\mathbf{i}	\mathbf{i}	\mathbf{t}	\mathbf{i}	\mathbf{f}	\mathbf{f}	\mathbf{t}	\mathbf{f}	\mathbf{i}	\mathbf{i}
\mathbf{t}	\mathbf{f}	\mathbf{u}	\mathbf{i}	\mathbf{t}	\mathbf{f}	\mathbf{f}	\mathbf{t}	\mathbf{t}	\mathbf{f}							

From now on we will sometimes use the convention that:

$$\neg(\neg\ell) \text{ is identified with } \ell. \quad (3)$$

Definition 2.2 We shall say that a set L of literals is *closed under negation* if for every $\ell \in L$, also $\neg\ell \in L$, where convention (3) applies. □

A discussion of \rightarrow can still be helpful. Observe that implication can only be \mathbf{t} or \mathbf{f} . It reflects the following principles:

- truth or falsity of a given proposition can only be deduced on the basis of true premises;
- on the basis of inconsistent premises one should only derive inconsistent conclusions;
- one should not derive conclusions on the basis of false or unknown premises.

Let us extend the semantics to full first-order language. As we deal with databases, in the rest of the paper we assume that interpretations and sets of constants are finite.

Definition 2.3 Let \mathcal{C} be a set of constants and v be a *valuation* assigning constants of \mathcal{C} to variables. By an *interpretation over \mathcal{C}* we mean any set \mathcal{I} of ground literals such that for any constant c occurring in literals in \mathcal{I} , $c \in \mathcal{C}$. The *truth value of formula A in interpretation \mathcal{I} under valuation v*, denoted by $\mathcal{I}^v(A)$, is defined by:⁴

⁴Note that $\mathcal{I}(v(A))$ has been introduced in Definition 2.1.

$$\begin{aligned}
 \mathcal{I}^v(\ell) &\stackrel{\text{def}}{=} \mathcal{I}(v(\ell)), \text{ where } \ell \text{ is a literal;} \\
 \mathcal{I}^v(\neg B) &\stackrel{\text{def}}{=} \neg \mathcal{I}^v(B); \\
 \mathcal{I}^v(B \circ C) &\stackrel{\text{def}}{=} \mathcal{I}^v(B) \circ \mathcal{I}^v(C) \text{ for } \circ \in \{\wedge, \vee, \rightarrow\}; \\
 \mathcal{I}^v(\forall X B(X)) &\stackrel{\text{def}}{=} \min\{\mathcal{I}^v(B(c)) \mid c \in \mathcal{C}\}; \\
 \mathcal{I}^v(\exists X B(X)) &\stackrel{\text{def}}{=} \max\{\mathcal{I}^v(B(c)) \mid c \in \mathcal{C}\},
 \end{aligned}$$

where the meaning of \circ at right-hand sides of equalities is provided in Table 1 and \min , \max are operations of minimum and maximum w.r.t. ordering (2).

We say an interpretation \mathcal{I} over \mathcal{C} is a *model of a formula A*, if for any valuation v assigning constants of \mathcal{C} to variables, $\mathcal{I}^v(A) = \mathbf{t}$. \triangleleft

3. The 4QL* Query language

As 4QL* extends 4QL, let us first recall some basic aspects of 4QL. As the underpinning principle, openness of the world is assumed, which may lead to the lack of information. Negative conclusions may lead to inconsistencies. To reduce the unknown/inconsistent zones, *modules* and *external literals* provide means for:

- the application-specific disambiguation of inconsistent information;
- the use of Local Closed World Assumption (thus also Closed World Assumption, whenever needed);
- the implementation of various forms of nonmonotonic and defeasible reasoning.

For related material see 4ql.org, where one can also find experimental open-source interpreters of 4QL.

For better readability we sometimes write ‘*expr*’ (in apostrophes) to denote *expr*.

Definition 3.1

- By a *rule* we mean any expression of the form:

$$\ell :- b_{11}, \dots, b_{1i_1} \mid \dots \mid b_{m1}, \dots, b_{mi_m}. \quad (5)$$

where:

- * ℓ is a literal, called the *conclusion* of the rule, denoted further by $\text{concl}(\varrho)$, and the expression at the right-hand side of ‘ $:-$ ’ is called the *antecedent* (and also the *set of premises*) of the rule;
- * $\{b_{11}, \dots, b_{1i_1}, \dots, b_{m1}, \dots, b_{mi_m}\}$ is a (possibly empty) set of literals;
- * ‘ $,$ ’ and ‘ $|$ ’ abbreviate conjunction and disjunction, respectively;
- * every variable appearing in ℓ appears also in each conjunction b_{k1}, \dots, b_{ki_k} ($1 \leq k \leq m$) of the rule.
- A rule without variables is called a *ground rule*. A ground rule with the empty antecedent is called a *fact*, where it is assumed that rule ‘ $\ell :- .$ ’ is understood as ‘ $\ell :- \mathbf{t}.$ ’ (For facts we further use notation ‘ $\ell.$ ’ rather than ‘ $\ell :- .$ ’)
- A 4QL* program is a finite set of rules. \triangleleft

Definition 3.2 Let S be a set of rules and let \mathcal{C} be the set of constants occurring in S . We say that an interpretation \mathcal{I} over \mathcal{C} is a *model* of S iff for each rule ' $\ell :- \beta.$ ' in S , the interpretation \mathcal{I} is a model of $(\beta \rightarrow \ell)$. \triangleleft

As we have indicated, 4QL* allows multiple rules with the same conclusion. Such extension of 4QL has been discussed in [6] but no algorithm for computing well-supported models dealing with this case has been provided there.

To illustrate the intended semantics, consider the following two rules:

$$\ell :- \beta_1. \quad (6)$$

$$\ell :- \beta_2. \quad (7)$$

As observed in [6], a formal meaning of rules with the same conclusion is given by:

$$((\beta_1 \rightarrow \ell) \wedge (\beta_2 \rightarrow \ell)) = ((\beta_1 \tilde{\vee} \beta_2) \rightarrow \ell), \quad (8)$$

where $=$ denotes equality on truth values of formulas and $\tilde{\vee}$ stands for the disjunction w.r.t. ordering:

$$u < f < t < i. \quad (9)$$

That is, $u \tilde{\vee} w \stackrel{\text{def}}{=} \max' \{u, w\}$, where \max' is the maximum w.r.t. (9). The following example illustrates the difference between \vee and $\tilde{\vee}$.

Example 3.3 Consider the following rules related to scenario (1) of Section 1:

$$\begin{aligned} jail(X) &:- \neg hasAlibi(X), weakEvidence(X). \\ jail(X) &:- evidence(X). \end{aligned} \quad (10)$$

These rules are equivalent to:

$$((\neg hasAlibi(X) \wedge weakEvidence(X)) \tilde{\vee} evidence(X)) \rightarrow jail(X). \quad (11)$$

If the value of $(\neg hasAlibi(X) \wedge weakEvidence(X))$ is t and the value of $evidence(X)$ is i , then the disjunction in the antecedent of implication (11) thus also of its conclusion, $jail(X)$, evaluates to i , which makes a perfect intuitive sense. With $\tilde{\vee}$ replaced by \vee the result of the antecedent as well as of $jail(X)$ is t , which is not what one really wants in such a case.

On the other hand, if we consider $(\neg evidence(X) \vee hasAlibi(X)) \rightarrow freed(X)$ then with $\neg evidence(X)$ being t and $hasAlibi(X)$ being i we have $freed(X)$ being t , which is intuitively better in this case than i obtained by replacing \vee by $\tilde{\vee}$. \triangleleft

4. What are Well-Supported Models?

The semantics of 4QL* is defined by well-supported models sharing the intuitions of [1, 2]. Namely, querying a 4QL* program depends on querying the well-supported model of this program. Intuitively, a model is *well-supported* if all derived literals are supported

by a reasoning grounded in facts. In the rest of the paper “well-supported models” will be referred to as *ws-models*.

To simplify presentation, in the current section we assume that all literals are ground (no variables allowed – see Definition 2.1). In particular, rather than allowing rules with variables, we consider their ground instances. Assuming that domains are restricted to constants appearing in rules, ground instances of rules define the same set of models as rules. Therefore, all formulated definitions and results can easily be extended to rules containing variables.

Example 4.1 The following table presents exemplary rules and their ground instances.

Rules	Ground instances
$p(X) :- q(X) \mid r(X).$	$p(a) :- q(a) \mid r(a).$
$r(b) :- q(X), s(X).$	$p(b) :- q(b) \mid r(b).$
$q(a).$	$r(b) :- q(a), s(a).$
	$r(b) :- q(b), s(b).$
	$q(a).$

That is, ground instances are obtained by substituting variables by constants appearing in rules. Observe that the process of grounding rules may result in multiple rules with the same conclusion. \triangleleft

To illustrate the idea of ws-models let us start with an example considered in [1].

Example 4.2 Let S be the following set of rules:

$$\begin{aligned} w &:- o \mid r . \\ r &:- w . \\ \neg o &:- r . \\ o . \end{aligned}$$

A minimal model of S is $\{o, \neg o, w, r\}$. However, the only fact, o , has in this model the value f so there are no facts supporting the truth of w and r in this model. The intuitively correct model for S is $\{o, \neg o, w, \neg w, r, \neg r\}$, where inconsistency of w and r is concluded from the inconsistency of o . \triangleleft

We also require that the reasoning starts from facts [1,2]. Otherwise it may be possible to find minimal models, where some true literals are not consequences of facts, as illustrated by the following example.

Example 4.3 Let S consist of rules:

$$\begin{aligned} w &:- o \mid r . \\ \neg o . \\ o . \end{aligned}$$

A minimal set of literals satisfying S is $\{o, \neg o, r, w\}$ but neither r nor w follow from facts, as the only fact, o , is inconsistent in this model. Another minimal model of S is $\{o, \neg o, w, \neg w\}$ and this is the intended model. \triangleleft

The minimal model of Example 4.2 can be obtained by DATALOG-like evaluation [8] with negative literals substituted by fresh positive literals. The following definition formally introduces such substitution.

Definition 4.4 Let S be a set of (ground) rules and L be a set of literals. If $\neg\ell$ is a negative literal then by its *duplicate* we understand a fresh literal representing its positive part, for simplicity denoted by ℓ' .

By $Pos(S)$ (respectively, $Pos(L)$) we understand the DATALOG program (respectively, the set of literals) obtained from S (respectively, from L) by replacing each negative literal $\neg\ell$ by its duplicate ℓ' . \triangleleft

The result of DATALOG-like evaluation can now be defined as follows.

Definition 4.5 Let S be a set of (ground) rules. Then gen^S is obtained from the least model of $Pos(S)$ by substituting literals of the form ℓ' with $\neg\ell$. \triangleleft

Since gen^S is defined by means of a DATALOG program, the required least model obviously exists (see, e.g., [8]).

Proposition 4.6 For any S set of (ground) rules and any model L of S , $gen^S \subseteq L$.

Proof: Observe that gen^S is obtained from the least Herbrand model of $Pos(S)$, so it contains literals which are in every model of $Pos(S)$. The replacement of literals ℓ' by $\neg\ell$ can only make some literals inconsistent, so can only increase the set of consequences. \triangleleft

It should be noted that gen^S need not be a model of S : for S of Example 4.3 we have $gen^S = \{o, \neg o, w\}$ which is not a model. Therefore we have to correct gen^S by adding inconsistent literals obtained by firing rules with inconsistent antecedents. On the other hand, the minimal model of Example 4.2 is gen^S , but in this model w, r are concluded on the basis of temporarily true premise o which later became inconsistent. The DATALOG-like reasoning should then be repeated after retracting inconsistent premises and its result should be corrected. The retraction can be done by removing the rules whose conclusions became inconsistent in the previous stage of reasoning. The rules with inconsistent conclusions are always satisfied and need not participate in the DATALOG-like reasoning.

The consequence operator, addressing these phenomena can be defined by means of three stages, assuming that all consequences are grounded in facts:

– generate the least set of conclusions by DATALOG-like reasoning; (12)

– retract conclusions based on defeated premises; (13)

– correct (minimally) the obtained set of literals to make it a model. (14)

These stages have to be repeated until no further retractions are needed, obtaining the ws-model since all conclusions are supported by facts and no conclusion is obtained using an assumption defeated later.

We now define formally such a consequence operator. The first stage is based on the already defined gen^S operator. The remaining stages deal with retraction of defeated conclusions and inconsistency propagation. For their formal definition we need some auxiliary notions and results presented below.

Proposition 4.7 Let S be a set of (ground) rules and:

$$\mathbb{N}(S) \stackrel{\text{def}}{=} \{L \mid L \text{ is a set of ground literals occurring in } S, \text{ and is closed under negation}\}.$$

Then set inclusion \subseteq is a complete partial order on $\mathbb{N}(S)$.

Proof: Proposition 4.7 can easily be justified by observing that:

for any sets $X, Y \in \mathbb{N}(S)$, both $(X \cup Y) \in \mathbb{N}(S)$ and $(X \cap Y) \in \mathbb{N}(S)$.

Extending the reasoning to arbitrary unions and intersections of sets in $\mathbb{N}(S)$ is then straightforward. \triangleleft

The following operator reflects the process of inconsistency propagation.

Definition 4.8 Let S be a set of (ground) rules and X, Y be sets of literals. We define:

$$\begin{aligned} \delta_X^S(Y) &\stackrel{\text{def}}{=} Y \cup \{\ell, \neg\ell \mid \text{there is a rule } ' \ell :- \beta.' \in S \\ &\quad \text{such that } (X \cup Y)(\beta) = \mathbf{i} \text{ and } (X \cup Y)(\ell) \neq \mathbf{i}\}. \end{aligned} \quad (15)$$

We have the following proposition.

Proposition 4.9 For any set of (ground) rules S and any set of ground literals X , δ_X^S is monotone on $\mathbb{N}(S)$.

Proof: Let $Y, Z \in \mathbb{N}(S)$ such that $Y \subseteq Z$. Note that in rules occurring in (15), true literals of $(X \cup Y)$ as well as of $(X \cup Z)$ have to appear in X , which is fixed.

Let $\ell \in \delta_X^S(Y)$. If $\ell \in Y$ then also $\ell \in Z \subseteq \delta_X^S(Z)$. If $\ell \notin Y$ then there is a rule ' $\ell :- \beta.$ ' $\in S$ or ' $\neg\ell :- \beta.$ ' $\in S$ such that $(X \cup Y)(\beta) = \mathbf{i}$. The same rule satisfies $(X \cup Z)(\beta) = \mathbf{i}$, so both $\ell, \neg\ell \in \delta_X^S(Z)$, in particular $\ell \in \delta_X^S(Z)$. \triangleleft

Of course, for any set of ground literals X and any $Y \in \mathbb{N}(S)$, also $\delta_X^S(Y) \in \mathbb{N}(S)$. Therefore, by Propositions 4.7 and 4.9, δ_X^S has the least fixpoint on $\mathbb{N}(S)$, i.e., there is the smallest set Y such that $\delta_X^S(Y) = Y$.

Definition 4.10 By $\Delta^S(X)$ we understand the least fixpoint of δ_X^S on $\mathbb{N}(S)$. \triangleleft

In what follows we shall use the following notation, where X is a set of ground literals: $true(X) \stackrel{\text{def}}{=} \{\ell \mid X(\ell) = \mathbf{t}\}$ and $incons(X) \stackrel{\text{def}}{=} \{\ell, \neg\ell \mid X(\ell) = \mathbf{i}\}$. Of course, for any such set X ,

$$X = true(X) \cup incons(X). \quad (16)$$

The following definition introduces operator Pre^S which formalizes the stages of reasoning (12)–(14) and is crucial in our definition of ws-models.

Definition 4.11 Let S be a set of (ground) rules. By the set of *pre-consequences of a set of literals X of S* , denoted by $\text{Pre}^S(X)$, we understand the set of literals defined by:

$$\text{Pre}^S(X) \stackrel{\text{def}}{=} \text{correct}^S(\underbrace{\text{incons}(X) \cup \text{gen}^T(X)}_{\text{retract (formalizing (13))}}), \quad (17)$$

where:

- $\text{correct}^S(X) \stackrel{\text{def}}{=} X \cup \Delta^S(X);$
- $T(X) \stackrel{\text{def}}{=} S - \{\varrho \in S \mid \text{concl}(\varrho) \in \text{incons}(X)\}.$ \triangleleft

Since Δ^S is well-defined, Pre^S is well-defined, too.

Of course, for any set L of ground literals,

$$\text{Pre}^S(L) = \text{incons}(L) \cup \text{gen}^T(L) \cup \Delta^S(\text{incons}(L) \cup \text{gen}^T(L)). \quad (18)$$

Proposition 4.12 Let S be a set of (ground) rules. For any set of literals L , $\text{Pre}^S(L)$ is a model of S .

Proof: We use characterization (18):

- rules with conclusions in $\text{incons}(L)$ are always satisfied;
- rules with antecedents inconsistent in $\text{Pre}^S(L)$ whose conclusions are not inconsistent are corrected by Δ^S . Since Δ^S only makes conclusions inconsistent, respective rules become satisfied;
- rules with antecedents true in $\text{Pre}^S(L)$ have their conclusions in $\text{gen}^T(L)$, so are satisfied by true conclusions;
- rules with antecedents unknown or false in $\text{Pre}^S(L)$ are always satisfied. \triangleleft

In general, operator Pre^S is not monotone. For example, if $S = \{p :- q \mid r.\}$ then:

$$\text{Pre}^S(\{q, \neg q\}) = \{q, \neg q, p, \neg p\} \not\subseteq \{q, \neg q, p\} = \text{Pre}^S(\{q, \neg q, r\}).$$

However, as we will show, it has a fixpoint reflecting our ideas (12)–(14) concerning construction of ws-models. We will use the following standard notation, where X is a set of ground literals and $i > 0$:

$$(\text{Pre}^S)^i(X) \stackrel{\text{def}}{=} \begin{cases} \text{Pre}^S(X) & \text{when } i = 1; \\ \text{Pre}^S((\text{Pre}^S)^{i-1}(X)) & \text{when } i > 1. \end{cases}$$

Note that, by Proposition 4.12, for $i > 0$: $(\text{Pre}^S)^i(\text{gen}^S)$ is a model of S . Moreover, as we will show (see Proposition 5.4), for $i > 0$, $(\text{Pre}^S)^i(\text{gen}^S) \subseteq (\text{Pre}^S)^{i+1}(\text{gen}^S)$. We then have the following definition.

Definition 4.13 For any set S of (ground) rules, the well supported model of S , denoted by W^S , is defined by:

$$W^S = \bigcup_{i>0} (\text{Pre}^S)^i(\text{gen}^S). \quad (19)$$

\triangleleft

5. Computing Well-Supported Models

Let us now prove that ws-models can be computed in deterministic polynomial time. We shall need the following lemma and proposition.

Lemma 5.1 Let S be a set of rules. Then for arbitrary set L of ground literals which is a model of S and such that $true(L) \subseteq true(gen^S)$, we have $L \subseteq Pre^S(L)$.

Proof: Let $\ell \in L$. Consider $Pre^S(L)$. Recall characterization (18):

$$Pre^S(L) = incons(L) \cup gen^{T(L)} \cup \Delta^S(incons(L) \cup gen^{T(L)}).$$

If $\ell \in incons(L)$ then $\ell \in Pre^S(L)$.

Assume $\ell \notin incons(L)$. By (16), from $\ell \in L$ and $\ell \notin incons(L)$ we have that $\ell \in true(L)$. By Proposition 4.12, $Pre^S(L)$ is a model of S . According to Proposition 4.6, in such a case $gen^S \subseteq Pre^S(L)$. By assumption, $true(L) \subseteq true(gen^S)$, so we have:

$$true(L) \subseteq true(gen^S) \subseteq gen^S \subseteq Pre^S(L).$$

As $\ell \in true(L)$, we conclude that $\ell \in Pre^S(L)$. \triangleleft

Since Pre^S can only transform true literals into inconsistent ones, we have the following proposition.

Proposition 5.2 Let S be a set of rules and L be a set of ground literals such that $true(L) \subseteq true(gen^S)$. Then $true(Pre^S(L)) \subseteq true(gen^S)$, too. \triangleleft

Corollary 5.3 Let S be a set of rules. Then:

$$\text{for } i > 0: true((Pre^S)^i(gen^S)) \subseteq true(gen^S). \quad (20)$$

Proof: We proceed by induction on i .

For $i = 1$: from $true(gen^S) \subseteq true(gen^S)$, by Proposition 5.2, we have that

$$true(Pre^S(gen^S)) \subseteq true(gen^S).$$

Assume that our corollary holds for $i > 0$, i.e., $true((Pre^S)^i(gen^S)) \subseteq true(gen^S)$. Then, again by Proposition 5.2, we conclude that:

$$true(Pre^S((Pre^S)^i(gen^S))) \subseteq true(gen^S). \quad \triangleleft$$

Proposition 5.4 Let S be a set of rules. Then:

$$\text{for } i > 0: (Pre^S)^i(gen^S) \subseteq (Pre^S)^{i+1}(gen^S). \quad (21)$$

Proof: By Proposition 4.12, for $i > 0$, $(Pre^S)^i(gen^S)$ is a model of S . By Corollary 5.3, $true((Pre^S)^i(gen^S)) \subseteq true(gen^S)$ so, by Lemma 5.1, we conclude that $(Pre^S)^i(gen^S) \subseteq Pre^S((Pre^S)^i(gen^S)) = (Pre^S)^{i+1}(gen^S)$. \triangleleft

Function generate(S : set_of_ground_rules): interpretation;

```

/* computes the least model of  $Pos(S)$  */  

/* (naive bottom up evaluation - see [8]) */  

/* and returns the corresponding four-valued */  

/* interpretation  $gen^S$  of Definition 4.5 */  

1 set  $\mathcal{I} = \emptyset$ ;  

2 while there is a rule ' $\ell :- \beta.$ '  $\in Pos(S)$  such that  $\mathcal{I}(\beta) = \text{t}$  and  $\mathcal{I}(\ell) \neq \text{t}$  do  

3   set  $\mathcal{I} = \mathcal{I} \cup \{\ell\}$ ;  

4 return  $\{\ell \mid \ell \in \mathcal{I} \text{ and } \ell \text{ is not primed}\} \cup \{\neg\ell \mid \ell' \in \mathcal{I}\}$ ;
```

Function findCorrection(S : set_of_ground_rules, \mathcal{I} : interpretation): interpretation;

```

/* computes the set  $\Delta^S$  of Definition 4.10. */  

1 set  $\Delta = \emptyset$ ;  

2 set  $\Gamma = \mathcal{I}$ ;  

3 while there is a rule ' $\ell :- \beta.$ '  $\in S$  such that  $\Gamma(\beta) = \text{i}$  and  $\Gamma(\ell) \neq \text{i}$  do  

4   set  $\Delta = \Delta \cup \{\ell, \neg\ell\}$ ;  

5   set  $\Gamma = \Gamma \cup \{\ell, \neg\ell\}$ ;  

6 return  $\Delta$ ;
```

Algorithm 1: Computing well-supported models for 4QL* programs.**Input:** A set of ground rules S .**Output:** The unique well-supported model \mathcal{I}^S for S .

```

/* computes the ws-model of  $S$  by applying */  

/* its definition (19) together with */  

/* characterization (18) of  $Pre^S$  */  

1 set  $\mathcal{I}^S = \text{generate}(S)$ ;  

2 repeat  

3   set  $\mathcal{I} = \mathcal{I}^S$ ;  

4   set  $\mathcal{I}^S = \text{incons}(\mathcal{I}^S) \cup \text{generate}(S - \{\varrho \mid \text{concl}(\varrho) \in \text{incons}(\mathcal{I}^S)\})$ ;  

5   set  $\mathcal{I}^S = \mathcal{I}^S \cup \text{findCorrection}(S, \mathcal{I}^S)$ ;  

6 until  $\mathcal{I}^S = \mathcal{I}$ ;
```

Therefore each iteration $(Pre^S)^i(gen^S)$ either adds new literals or a fixpoint is reached. Since only a polynomial number of literals can be added, the fixpoint is reached after a polynomial number of steps. We then have the following corollary.

Corollary 5.5 For any set of (ground) rules S , its well-supported model is determined uniquely and can be computed in deterministic polynomial time. \triangleleft

By characterization (19), to compute the well-supported model one can iterate $Pre^S(X)$ starting with $X = gen^S$. Algorithm 1 reflects such computations. It uses:

- function *generate*, implementing gen^S ;
- function *findCorrection*, implementing Δ^S of Definition 4.10.

The algorithm is presented in its simplest form. Of course, in actual implementations it should be optimized, e.g., adapting known optimizations for SQL and DATALOG queries (see, e.g., [8,9]).

Since Algorithm 1 applies Definition 4.13 and computes ws-models directly using (19), its correctness is obvious.

Remark 5.6

- If there are no inconsistencies, Algorithm 1 terminates after one iteration of the **repeat** loop. In fact, in such cases a single application of function ‘generate’ suffices. Therefore, when there are no inconsistencies, algorithms designed for DATALOG can be adapted almost directly.
- Algorithm 1 indicates that inconsistencies are more computationally demanding than lack of information. It is then better to resolve them as early as possible. Also, when a particular set of facts or relations can be inconsistent, it is generally a good idea to isolate them, together with their smallest context, in separate modules.
- Observe that the number of iterations of the **repeat** loop of Algorithm 1 is typically rather small (often one or two iterations suffice). \triangleleft

6. Modules and External Literals

To express nonmonotonic/defeasible rules we need the concept of modules and external literals. The concept of modules is known from many programming languages as an encapsulation tool. In 4QL a layered architecture is required.⁵ A *layer* is a set of modules. We also assume a partial order \preceq on modules, where $m \preceq n$ means that m is in a layer lower than n . In 4QL* we adopt the same architecture. This architecture is implicit: a 4QL* program is well-formed if there are suitable layers and \preceq satisfying the following definition.

⁵The layered architecture can be relaxed as in [6].

Definition 6.1

- An *external literal* is an expression of one of the forms:

$$A.R, \neg A.R, A.R \text{ IN } T, \neg A.R \text{ IN } T,$$

where A is a module name, R is a positive literal and $T \subseteq \{\mathbf{f}, \mathbf{u}, \mathbf{i}, \mathbf{t}\}$.⁶ A is called the *reference module* of the external literal.

- An external literal may only appear in rule antecedents of a module B , provided that its reference module is in a layer strictly lower w.r.t. \preceq than B .
- We write $\ell = v$ to stand for $\ell \text{ IN } \{v\}$. \triangleleft

The layered architecture allows us to compute well-supported models starting from the basic modules (at the lowest layer) and proceeding with modules level by level using Algorithm 1. This way values of external literals are computed before they are accessed.

To illustrate and motivate the architecture let us formalize scenario (1) discussed in Section 1.

Example 6.2 Scenario (1) can be formalized using two modules: *trial* and *investigation*. Module *investigation* can consist of rules:

$$\begin{aligned} \text{guilty}(X) &:- \text{evidence}(X). \\ \neg \text{guilty}(X) &:- \text{hasAlibi}(X). \end{aligned}$$

It can also contain facts about collected evidences and alibis.

Note that for particular instances of X we can have the following cases:

- both $\text{evidence}(X)$ and $\text{hasAlibi}(X)$ is \mathbf{f} or \mathbf{u} , in which case $\text{guilty}(X)$ is unknown;
- $\text{evidence}(X)$ is \mathbf{t} and $\text{hasAlibi}(X)$ is in $\{\mathbf{f}, \mathbf{u}\}$, then $\text{guilty}(X)$ becomes \mathbf{t} ;
- $\text{evidence}(X)$ is in $\{\mathbf{f}, \mathbf{u}\}$ and $\text{hasAlibi}(X)$ is \mathbf{t} , then $\text{guilty}(X)$ becomes \mathbf{f} ;
- both $\text{evidence}(X)$ and $\text{hasAlibi}(X)$ is \mathbf{t} , then $\text{guilty}(X)$ becomes \mathbf{i} ;
- $\text{evidence}(X)$ is \mathbf{i} or $\text{hasAlibi}(X)$ is \mathbf{i} , then $\text{guilty}(X)$ becomes \mathbf{i} .

Module *trial* can consist of rules:

$$\begin{aligned} \text{jail}(X) &:- \text{investigation.guilty}(X) = \mathbf{t}. \\ \text{freed}(X) &:- \text{investigation.guilty}(X) \in \{\mathbf{f}, \mathbf{u}\}. \\ \text{furtherInvestigation}(X) &:- \text{investigation.guilty}(X) = \mathbf{i}. \end{aligned}$$

Of course, module *investigation* can be placed in a layer lower than *trial*.

Note also that module *trial* can have rules resolving all or some cases differently. For example, rules:

$$\begin{aligned} \text{jail}(X) &:- \text{investigation.guilty}(X) = \mathbf{t} | \\ &\quad \text{investigation.evidence}(X) = \mathbf{t}, \text{investigation.hasAlibi}(X) = \mathbf{i}. \\ \text{freed}(X) &:- \text{investigation.guilty}(X) \in \{\mathbf{f}, \mathbf{u}\} | \\ &\quad \text{investigation.evidence}(X) = \mathbf{i}, \text{investigation.hasAlibi}(X) = \mathbf{t}. \\ \text{furtherInvestigation}(X) &:- \text{investigation.evidence}(X) = \mathbf{i}, \\ &\quad \text{investigation.hasAlibi}(X) = \mathbf{i}. \end{aligned}$$

⁶The intended meaning of $A.R \text{ IN } T$ is that the truth value of $A.R$ is in the set T . If R is not defined in the module A then the value of $A.R$ is \mathbf{u} .

define another reasonable policy of resolving gaps in beliefs and potential inconsistencies. \triangleleft

The above example illustrates one of the principles underlying 4QL*:

reasoning with unknown/inconsistent information is possible by encapsulating beliefs in modules and completing information/resolving inconsistencies is done in modules of a higher level.

Let us also emphasize the following important property of modules and external literals.

Theorem 6.3 4QL* with modules and external literals captures PTIME over ordered databases, i.e., every query computable in deterministic polynomial time w.r.t. size of the database domain, is expressible in 4QL* with modules and external literals, assuming that a linear order on the database domain is available. \triangleleft

The proof of this theorem directly follows from a similar result for 4QL proved in [2], since 4QL* is an extension of 4QL.

7. Conclusions

In the paper we have discussed 4QL*, a tool for expressing agent's belief bases allowing for incomplete and inconsistent information. 4QL* is simple yet powerful in expressing lightweight forms of nonmonotonic and inconsistency resolution rules. It enjoys an intuitive semantics and is tractable w.r.t. data complexity and captures all tractable queries.

We have proposed a simplified and more general definition of well-supported models and a novel algorithm for computing such models. As literals included in well-supported models are well justified, the proposed solution, together with ideas of [3,4], provides a firm theoretical foundation for specifying agents' beliefs and reasoning about them.

Acknowledgment

We would like to thank Miłosz Pacholczyk for helpful comments.

References

- [1] J. Małuszyński and A. Szałas, "Living with inconsistency and taming nonmonotonicity," in *Datalog 2.0* (O. de Moor, G. Gottlob, T. Furche, and A. Sellers, eds.), vol. 6702 of *LNCS*, pp. 334–398, Springer-Verlag, 2011.
- [2] J. Małuszyński and A. Szałas, "Logical foundations and complexity of 4QL, a query language with unrestricted negation," *Journal of Applied Non-Classical Logics*, vol. 21, no. 2, pp. 211–232, 2011.
- [3] B. Dunin-Kęplicz and A. Szałas, "Epistemic profiles and belief structures," in *KES-AMSTA* (G. Jezic, M. Kusek, N. Nguyen, R. Howlett, and L. Jain, eds.), vol. 7327 of *LNCS*, pp. 360–369, Springer-Verlag, 2012.
- [4] B. Dunin-Kęplicz and A. Szałas, "Distributed paraconsistent belief fusion," in *Intelligent Distributed Computing VI* (G. Fortino, C. Badica, M. Malgeri, and R. Unland, eds.), vol. 446 of *Studies in Computational Intelligence*, pp. 59–69, Springer-Verlag, 2013.

- [5] S. de Amo and M. Pais, "A paraconsistent logic approach for querying inconsistent databases," *International Journal of Approximate Reasoning*, vol. 46, pp. 366–386, 2007.
- [6] A. Szałas, "How an agent might think," *Logic J. IGPL*, 2013. To appear. DOI: 10.1093/jigpal/jzs051.
- [7] A. Vitória, J. Małuszyński, and A. Szałas, "Modeling and reasoning with paraconsistent rough sets," *Fundamenta Informaticae*, vol. 97, no. 4, pp. 405–438, 2009.
- [8] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley Pub. Co., 1996.
- [9] H. Garcia-Molina, J. Ullman, and J. Widom, *Database System Implementation*. Prentice-Hall, 2000.

Multi-Agent Systems Design and Implementation

Erlang as a High Performance Software Agent Platform

Wojciech TUREK¹

AGH University of Science and Technology, Krakow, Poland

Abstract. Despite many years of work on multi-agent platforms development, the problem of their performance remains unsolved. Even most mature solutions still suffer from significant limitations in terms of message-passing services and the number of agents which can work simultaneously. Large overheads are caused mostly by unsuitability of technology used for implementing the agent platform itself. In this paper several measures of an agent platform performance are proposed. Two popular agent platforms are thoroughly tested and compared to Erlang technology. The results of the experiments show that different approach to the problem of creating agent platform can give significantly better performance.

Keywords. multi-agent platform, Erlang technology

1. The Problem of Agents Performance

Agent paradigm has been widely studied over last few decades. Many different applications for agent systems have been tested and the paradigm definitely has proven its usability and significant advantages in many areas, including distributed computations [1], optimization [2] or control of multi-robot systems [3,4]. Although researchers do differ in the precise definition of agent, basic assumptions and concepts of the paradigm are commonly accepted [5,6]. This includes the most basic idea of **autonomy** of coexisting entities, which is based in the concept of Actor model proposed in the seventies [7]. The need of interactions between autonomous agents leads to another basic concept, which is **communication** based on asynchronous messages.

These two concepts form basic frames of agent paradigm. They also underlie the major advantages of the agent-based approach. Autonomy of elements of a system improves fault-tolerance – failure of a single element is not fatal for others. Well-separated components are easy to reuse in different systems. Separation of the problem into loosely related entities often solves the problem of scalability.

Autonomous agents, which can communicate using asynchronous messages, require proper environment of execution. Focusing on software agents, the environment can be defined as a computer system, which allows execution of agents threads or processes and supports mechanisms for transferring messages from one agent to another. The system is typically called an agent platform.

¹Corresponding Author: Wojciech Turek, AGH University of Science and Technology, 30 Mickiewicza Av. 30-059 Krakow, Poland; E-mail: wojciech.turek@agh.edu.pl

It is relatively easy to create an agent platform, therefore many different implementations have been developed over last decades. Although the basic features are common, the platforms can differ significantly in underlying technology and implemented services. Noble goal of standardization of services provided by agent platforms was the foundation of creation of the FIPA organization [8]. Commonly used standards, like FIPA Agent Communication Language (ACL), specify frames for development of agent platforms.

The variety of services provided by the most advanced agent platforms is impressive. However, there is one issue, which significantly reduces their usability in production applications: performance and limitations. The problem of agent platform performance has received significant attention over last years. A detailed review of research performed in this area can be found in [9]. The most general conclusion from these publications is that the performance of wide-spread used agent platform is unsatisfactory and that scalability is insufficient.

The performance of an agent-based system is determined by the performance of software agents and the performance of the agent platform. While the performance of agents is dependent on the problem solved and can be optimized by agents' creator, the performance of the agent platform creates impassable limits.

Agent platforms are typically implemented in popular imperative languages, using well-established technologies. This approach has many advantages. Existing libraries and components can be reused, platforms can be integrated with other systems and development of agents is not a hard task. However, an agent system could exhibit far better performance if the implementation was based on a technology, which natively supports actor model and message passing communication.

In this paper the performance of two popular agent platforms, namely JADE [10] and Magentix [11], is compared with the performance of the Erlang technology, which natively supports actor model and message-passing concurrency. In the following section a description of the tested platforms is presented. In the third section several measures, used for agent platform performance evaluation, are described and details on performed experiments and results are described.

Evaluation presented in this paper is focused on particular aspects only. It is focused on communication performance and the ability of hosting many agents simultaneously. Additional features supported by tested platforms, although often important, are not taken into consideration.

2. Analyzed Agent Platforms

Two popular agent platforms (JADE and Magentix) have been chosen to create a reference point for evaluating the performance of Erlang technology. Both platforms have been developed for many years now, and are considered advanced and mature solutions. The Erlang technology was originally created as a tool for programming telecommunication switches, however its features allow making comparisons with agent platforms.

2.1. JADE

Java Agent DEvelopment Framework (JADE) has been developed for more than 10 years. It is probably the most popular agent platform nowadays. It is written in Java, it

is also possible to compile JADE in .NET. JADE claims to be fully compliant with FIPA standards.

JADE agents are executed in JADE containers. A platform can interconnect several computers, each running a JADE container. There must be a selected, main container, which has to be started as first one. Main container is a single point of failure of the platform. Message passing mechanisms implemented in JADE are based on underlying network protocols.

2.2. Magentix

Magentix2 platform aims at providing high quality software, which could be used in industrial applications. It is also written in Java. It does not focus on full FIPA compliance, however it provides security services and tracing tools, which should help developing high quality programs.

Magentix2 uses several existing solutions. If provides a graphical user interface via the Tomcat [12] web server. By default Magentix2 uses Apache Qpid [13], which is a cross-platform messaging system. Qpid server also creates a single point of failure for the platform – all messages traffic passes through it.

2.3. Erlang

As mentioned before, Erlang is **not** an agent platform. At least its developers never called it so. Erlang [14] is a programming language designed by Ericsson for developing concurrent and distributed systems. It has been developed since early nineties, and became open-source in 1998. Fundamental assumptions that guided its creators were: high availability, fault tolerance, concurrency and communication, soft real-time and ease of maintenance. It resulted in creation of a functional programming language, which is being compiled to a byte code executed in a virtual machine.

The Erlang Virtual Machine can be run at almost every modern computer, embedded device or mobile phone. The EVMs can join together, creating a distributed runtime environment, where applications can almost transparently use all available hardware. In many applications this can be a crucial feature; for example each robot in a multi-robot system can host a node of a consistent agent environment on its on-board computer.

There is a very significant difference in the approach to creating a distributed runtime environment between Erlang and typical agent platforms. As shown in Figure 1 b), there is no message router in EVM network. Each node connects to all other known nodes, creating a complete network of connections.

This approach has two advantages. First of all, there is no single point of failure in the Erlang platform. Secondly, messages are routed directly to the recipient node, without passing through central router.

Erlang programs are composed of independent processes. Processes do never share common memory – the only way to communicate between processes is to send messages. The mechanism of sending messages is embedded in the language itself.

Erlang processes do not use operating system threads to realize concurrency. EVM implements own context switching. Results of this approach will be visible in tests of platform limits, presented in the following section.

It is easy to notice that the basic assumptions of the Erlang technology are very similar to those attributed to agent platforms. Although Erlang creators did not refer to

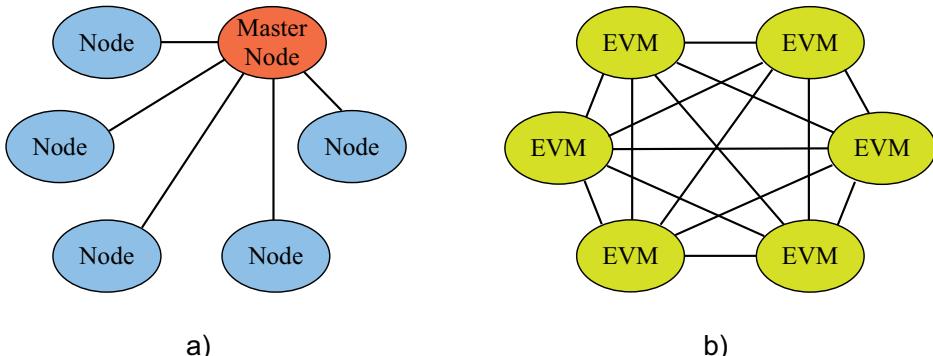


Figure 1. Different approach to connecting platform nodes. On the left typical approach with a selected routing node; on the right Erlang approach with complete network of connections.

the agent paradigm, Erlang language and runtime environment can definitely be used as an agent platform.

Erlang provides much more than the basic features mentioned so far. Several services of the EVM and the Erlang software development kit (called Erlang Open Telecom Platform, OTP) are very similar to more advanced services of typical agent platforms. A naming service is a good example of that. Each process has its unique identifier. However, each process can also register in platform naming service to become discoverable by any potential user.

Another very important feature of EVM is the mechanism of hot code swapping. Source code, executed by an agent can be changed on runtime, without stopping the agent. Even its internal state can be preserved during code modification.

The idea of using Erlang as an agent platform is not completely novel. Several years ago Di Stefano and Santoro published results of their work on an Erlang based agent platform, called eXAT [15]. They focused on providing advanced mechanisms for creating intelligent agents, like a complete rule based system. Unfortunately, the project seems to be discontinued now.

It is worth noticing that Erlang is not an experimental language. It is a widely used, industrial quality platform, which has proven its usability in many applications. Many large corporations use Erlang-based software for providing high-availability, large scale services.

The features of Erlang technology justify conducting performance test to verify, whether it can outperform popular and mature agent platforms. The results of the experiments are presented in the following section.

3. Performance Evaluation

Two basic assumptions of the agent paradigm, which were described in the first section, imply two most important responsibilities of an agent platform: managing of agents and providing mechanisms for asynchronous communication. The performance of providing these responsibilities can be measured and compared.

Most of the research in the area focuses on measuring messaging performance. A very interesting set of measures was proposed in [16] and later extended in [9]. The au-

thors used two types of agents: senders and repliers, which exchanged specified number of very small messages. Average round trip time was used as a measure of communication performance. The experiments showed that JADE platform outperforms two other tested solutions in terms of messaging performance.

Similar measure has been used in evaluation of messaging performance described in this paper, although particular scenarios are different. The experiments have been divided into two classes:

1. single computer with one agent container,
2. multiple computers with one container per computer.

All tests described in this section were conducted on identical computers, equipped with a four-core, 3GHz CPU, 4GB of RAM and a 32-bit operating system, Java 1.6 and Erlang 15.

3.1. Single Computer Messaging

Agent systems are often deployed on a single computer. A platform should support decent performance of passing messages between agents on a single node. This type of messaging does not require using network interface, therefore results should be much better than in case of distributed architecture.

The tests involved the same number of sender and replier agents. Each sender was supposed to send messages to randomly selected repliers, each time waiting for response. Senders sent 1000 messages of several bytes each. Figure 2 presents the results of the experiment.

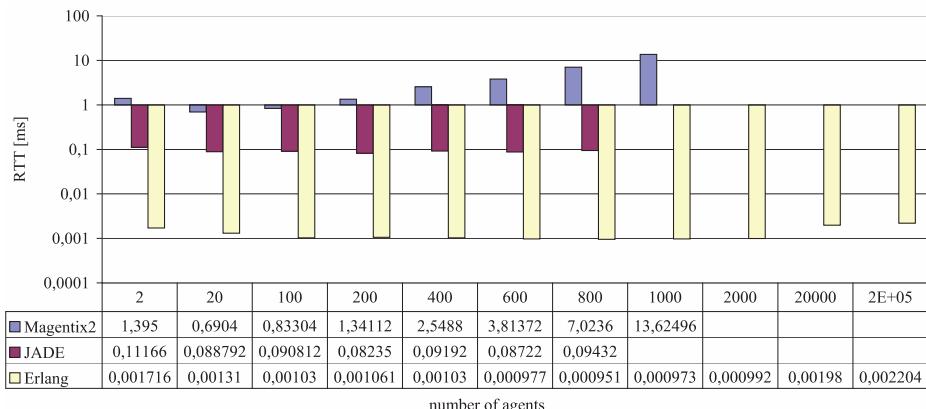


Figure 2. Results of single computer messaging test, where number of repliers was equal to the number of senders.

Proper implementation of messaging service should be able to deliver messages in a constant time, no matter how many messages are being sent. Results show that JADE and Erlang are able to keep the time on a constant level. Magentix platform failed to keep constant time. This is probably caused by use of external messaging service (Apache Qpid), which uses local network and also tends to scale very poorly. This will be also visible in following experiments.

Comparison of round trip time for tested platforms shows significant differences. Magentix is at least an order of magnitude slower than JADE, while Erlang is two orders of magnitude faster than JADE. This is an enormous difference of performance.

The table on the bottom of Figure 2 shows measured values in milliseconds. Missing values for Magentix and JADE platforms are caused by inability of finishing particular test. Typically the problem was caused by limitation on the maximum number of agents, which can be created in the platforms. The reason for this limitation lies in the way, the platforms implement agent. The platforms use native threads of the underlying operating system, which are rather heavy and cannot be created in thousands. Erlang implements its own, internal processes and scheduling mechanisms, which are much lighter. In this test the agent system was scaled up to 200000 agents. The evaluation of agent management capabilities of the platforms will be described later in this section.

3.2. Multiple Computers Messaging

This group of tests was conducted using clusters of 4, 8, 12, 16 and 20 identical computers connected with a 1Gb local area network. Number of agents on each node was fixed, because previous tests showed that for correct implementation average round trip time does not change significantly with the number of agents used.

The first test involved 10 senders and 10 repliers on each node. Each sender continuously sent messages to randomly selected remote replier, each time waiting for answers. The results of this series of experiments are presented in Figure 3.

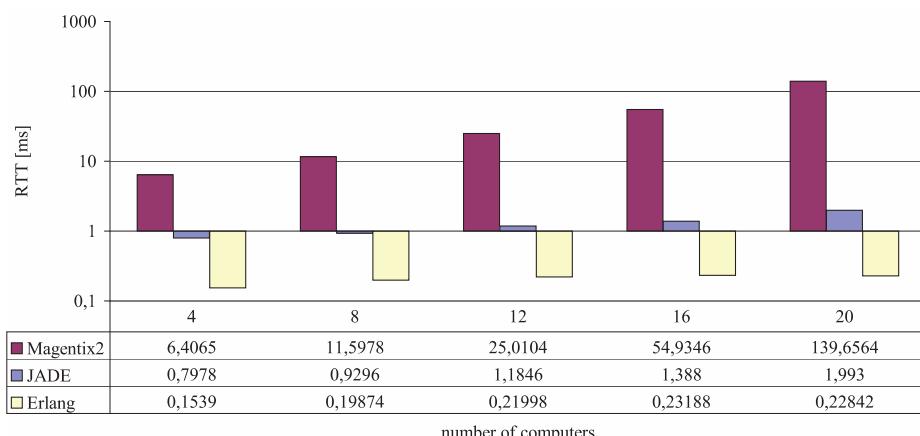


Figure 3. Results of multiple computers messaging test; number of repliers was equal to the number of senders.

The results show very important features of the analyzed platforms remote messaging services. Magentix uses centralized message broker (Qpid server), which routes messages between different computers. The results show, that the RTT (Round Trip Time) grows faster than linearly with the number of computers in the cluster. The values of RTT were huge compared to two other platforms.

JADE platform uses a centralized RMI (Remote Method Invocation) registry, which should be queried only once for location of particular container of the platform. After first

communication between agents in different containers the following messages should be sent directly to destination host. However the results show that the RTT grows linearly with the number of computers in the cluster, which is unexpected.

The solution used in Erlang directly sends messages to destination node using internal protocol based on TCP/IP. This results in almost constant RTT for 12, 16 and 20 nodes in the cluster, which is a desirable feature in terms of platform scalability.

Relation between performance of tested platforms is similar to the single computer scenarios. Magentix is at least an order of magnitude slower than JADE, while Erlang is about five times faster than JADE.

Performed experiments of messaging service show clearly that platforms do significantly differ in messaging performance. The most important conclusion is that Erlang provides much better performance in all cases and offers best scalability. The other two platforms failed to provide proper relation between RTT and growing load.

3.3. Agent Creation

As mentioned before, most of the research on agent platforms performance focus on messaging services. More basic responsibility of agent platform has received significant less attention.

Every agent platform must be able to host agents. Typically this includes at least operations of creation, execution and killing agents. The most important measures, characterizing the performance and limitations concerning hosting of agents, are:

1. number of agents, which can be hosted on a single node,
2. time of agent creation and startup.

The test concerning agent creation were conducted using single computer only. Idle agents were continuously created, but never performed any actions. In each step of the experiment 100 new agents were added. The results of the experiment are presented in Figure 4.

Magentix and JADE seem to provide almost constant time of agent creation in this scenario, since the first agent until reaching platform limit. Magentix is much slower than JADE in this aspect as well.

JADE platform managed to create between 800 and 900 agents before reaching the limit, Magentix performed slightly better. Both platforms refused to create more agents because of a lack of memory error, during thread creation.

In this test the most important advantage of Erlang appears clearly. Erlang does not use heavy operating system threads to implement concurrency. Its internal processes are lightweight and can be created in thousands without any problems. In the experiment, average time of agent creation did not exceed $0.1\mu s$ until tens of thousands of agents were present in the platform. Erlang managed to host 1000000 agents with no trouble.

4. Conclusions

Results of the tests described in this paper were a big surprise for the author. It turned out that three environments are hardly comparable in terms of messaging performance.

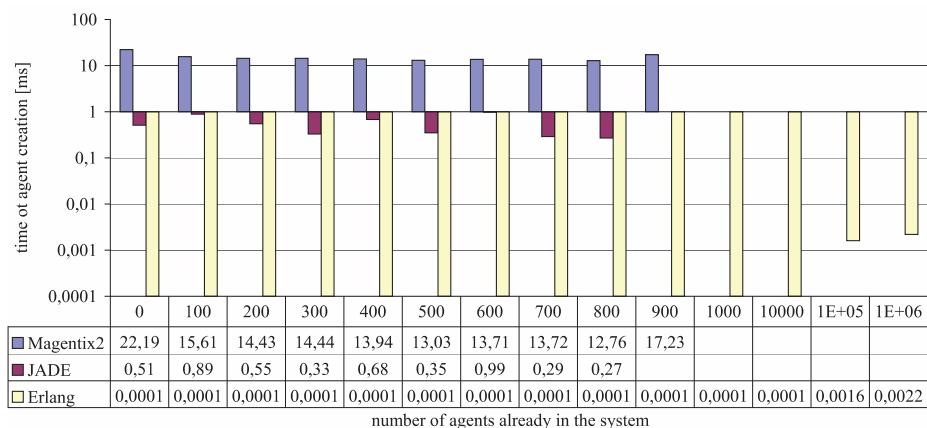


Figure 4. Results of agents creation tests where created agents remained idle.

The contest looks like a race between a hippopotamus, an elephant and a cheetah. The experiments suggest that even the best performing agent platforms suffer from extremely large overhead.

What is more, the Erlang platform presented significantly better scalability in terms of number of simultaneously working agents. Erlang was able to host 1000000 of working agents, which is three orders of magnitude better than platforms using native operating system threads.

The most important conclusion of this paper is that Erlang should be considered a very fast and flexible agent platform, or at least that it is a good basis for creating one. It can be successfully used for building large scale multi-agent systems. It is hardly possible that any platform written in popular imperative language could offer similar performance and scalability.

Further research will focus on more advanced features of agent platforms and methods for their implementation in Erlang. In particular agent-based planning algorithms, decision support methods and agent based simulation will be considered. This will require methods for efficient knowledge representation and processing [17], which can also be compared with existing platforms. The results will be compared with wider variety of existing platforms [18], including solutions dedicated for particular problems.

Acknowledgement

The research leading to this results has received funding from the Polish National Science Centre under the grant no. UMO-2011/01/D/ST6/06146.

References

- [1] A. Byrski, R. Schaefer. Formal Model for Agent-Based Asynchronous Evolutionary Computation. Proceedings of IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 78–85.
- [2] K. Cetnarowicz, R. Drezewski. Maintaining functional integrity in multi-agent systems for resource allocation. Computing and Informatics vol. 29, 2010, pp. 947-973.

- [3] W. Turek, K. Cetnarowicz, W. Zaborowski. Software Agent Systems for Improving Performance of Multi-Robot Groups. FUNDAMENTA INFORMATICA vol. 112 (1), 2011, pp. 103–117.
- [4] W. Turek. Extensible Multi-Robot System. Proceedings of the 8th International Conference on Computational Science, Springer LNCS vol. 5103, 2008, pp. 574–583.
- [5] K. Cetnarowicz, P. Gruer, V. Hilare. A Formal Specification of M-agent Architecture. Proceedings of the Multi-Agent Systems CEEMAS 2001, pp. 62–72.
- [6] K. Cetnarowicz. From Algorithm to Agent. Proceedings of the International Conference on Computational Science, Springer Verlag LNCS vol. 5545, 2009, pp. 825–834.
- [7] C. Hewitt, P. Bishop, R. Steiger. A universal modular ACTOR formalism for artificial intelligence. Proceedings of the 3rd Conference on Artificial Intelligence, Stanford, USA, 1973, pp. 235–245.
- [8] The Foundation for Intelligent Physical Agents. <http://www.fipa.org/> 2013.
- [9] J. Alberola, J. M. Such, A. Garcia-Fornes, A. Espinosa, V. Botti. A performance evaluation of three multiagent platforms. Artificial Intelligence Review 34(2), 2010, pp. 145–176.
- [10] F. Bellifemine, A. Poggi, G. Rimassa. JADE – A FIPA-compliant agent framework. Proceedings. of PAAM'99, London, UK, 1999, pp. 97–108.
- [11] J. M. Alberola, J. M. Such, A. Espinosa, V. Botti, A. Garca-Fornes. Scalable and Efficient Multiagent Platform Closer to the Operating System, Artificial Intelligence Research and Development, vol.184, 2008, pp. 7–15.
- [12] J. Brittain, I.F. Darwin. Tomcat: The Definitive Guide. O'Reilly Media, 2003.
- [13] J. Robie, C. Rolke, A. Young. Programming in Apache Qpid. Red Hat, Inc., 2011.
- [14] F. Cesarini, S. Thompson. Erlang Programming. A Concurrent Approach to Software Development. O'Reilly Media, 2009.
- [15] A. Di Stefano, C. Santoro. Supporting Agent Development in Erlang through the eXAT Platform. Software Agent-Based Applications, Platforms and Development Kits, 2005, pp. 47–71.
- [16] J. M. Such, J. M. Alberola, L. Mulet, A. Espinosa, A. Garcia-Fornes, V. Botti. Large-scale multi-agent platform benchmarks. In Languages, methodologies and Development tools for multi-agent systemS (LADS 2007). Proceedings of the Multi-Agent Logics, Languages, and Organisations - Federated Workshops, 2007, pp. 192–204.
- [17] S. Ernst, A. Ligeza. Knowledge Representation for Intelligent and Error-Prone Execution of Robust Granular Plans. A Conceptual Study. Proceedings of the International Florida Artificial Intelligence Research Society Conference, Florida, USA, 2009.
- [18] Ł. Faber, K. Pietak, A Byrski, M. Kisiel-Dorohinicki. Agent-based simulation in AgE framework. Advances in intelligent modelling and simulation: simulation tools and applications. Studies in Computational Intelligence, 2012, pp. 55-83.

Travel Routes Flexibility in Transport Systems

František ČAPKOVIČ^{a,1}

^a Institute of Informatics, Slovak Academy of Sciences, Slovakia

Abstract. Because of congestions in traffic a precomputed shortest route (path) of a vehicle from an initial location to a prescribed one cannot be always adhered. In such a case it is necessary to change it flexibly. However, to save cost the new route should be chosen in virtue of prescribed rules. The place/transition Petri nets, more precisely its simpler kind named as state machines (SM), is utilized here to model the real structure of the possible routes area. An algorithm for finding the least-cost firing sequence of SM transitions is proposed. To extend the approach on adjacent areas the agent-based approach is drawn too.

Keywords. agent, cooperation, least cost, place/transition Petri nets, shortest path, state machines, transport systems

1. Introduction

In modern society the traffic, and especially its successful control, has a high relevance. In the goods traffic (e.g. food-supply, logistics of different kinds, etc.) for shopping centers, factories, building industry (civil engineering) etc., the economy of the transportation is followed above all because losses/savings can be considerable, even huge. Therefore, methods for finding most suitable paths with least cost or the shortest time are investigated. Techniques depend on a kind of the transportation. For example, while at the distribution of foods, lively animals, the fluid concrete, at logistics for automobile factories etc., the shortest time is desirable first of all, elsewhere, e.g. at distribution of money in order to replenish cash in ATM (automatic teller machines) networks, primarily the safety of transport are expected. However, in both kinds the combination with finding the shortest path (to economize the fuel) is expected. Namely, unpredictable circumstances (congestions) occur in real situations. Traffic congestions are [1] the annoyance of almost all drivers, apart from large monetary losses, dangerous situations and damage to the environment. Thus, all participating parties will have benefits when congestion is reduced or fully eliminated. This might look easier than it seems, since causes are in large numbers and with varying sources. The US Department of Transportation performed a study into the main causes of congestion. They have two possible forms - recurring and non recurring. The former one contains: (i) physical bottlenecks (40%): insufficient road capacity, merge lanes at interchanges and bad roadway alignment; (ii) traffic control de-

¹Corresponding Author: František Čapkovič, Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9, 845 07 Bratislava, Slovakia; E-mail: Frantisek.Capkovic@savba.sk

vices (5%): ineffective management of capacity, due to poor signal timing for example. The latter one contains: (i) traffic incidents (25%): vehicle crashes and breakdowns; (ii) bad weather (15%): snow, fog, rain, and slipperiness for example; (iii) work zones (10%): can also be seen as temporary bottlenecks; (iv) special events (5%): other events like unexpected actuating demands.

The causes look very simple and mutually exclusive, but some event can cause other events to occur. One traffic incident might cause other traffic incidents or traffic jams on the other direction.

An unexpected change of the situation forces drivers to alter their originally planned routes and to adapt them to the actual traffic situation. However, there exists kinds of traffic when the willful changes are performed in order to avoid terroristic attacks - e.g. at the transport of money, prisoners, etc.

In [2],[3] an approach called Adaptive Gray Threshold Traffic Parameters Measurement (AGTTPM) for extracting traffic parameters such as traffic flow and average vehicle speed from real traffic data was proposed. This approach was based on an image processing algorithm and was able to obtain the parameters in real time that could be used for traffic control and traffic condition evaluation. The work [4] focuses on the problem of finding the routes with least travel time for a given transportation system modeled as a Petri net. Each transition in the net is associated with a cost that is related to the travel time computed from the traffic parameters obtained from the AGTTPM system. Given the starting location (the initial marking) and the destination (the final marking), the authors developed an algorithm that is able to obtain the routes (transition firing sequences) with the least travel time (the least total cost) from the starting point to the destination within a certain time period τ . In [5] the possibility of using hybrid Petri nets in transport systems is described too.

To study the corresponding problems simulation is frequently used. Among the simulation approaches, the agent-based simulation is utilized too. In this paper we will try to deal with the problem expressed by the following scenario: Consider a vehicle at a starting point and it can reach the destination via different travel routes. In general, when the theoretically shortest route is congested, the driver will take a longer route with less traffic so that to get the destination in shorter time compared with the congested route. We will develop an algorithm to obtain simultaneously both the shortest paths from a given initial state (the location in a road structure where the vehicle actually occurs) to a prescribed terminal state (the location where the driver wants to get) and the least-cost transition firing sequences efficiently. The place/transition Petri nets (P/T PN) [6] are used here to model the scenario.

2. Preliminaries

As to the structure P/T PN are bipartite directed graphs $\langle P, T, F, G \rangle$ with P, T, F, G being, respectively, the set of places, the set of transitions, the set of directed arcs from places to transitions and the set of directed arcs from transitions to places. Here, $P \cap T = \emptyset, F \cap G = \emptyset$. Moreover, P/T PN have their dynamics $\langle X, U, \delta, x_0 \rangle$ with X, U, δ, x_0 being, respectively, the set of states (marking the places), the set of discrete events (states of transitions), the transition function and the initial state vector. Here, $X \cap U = \emptyset, \delta : X \times U \rightarrow X$. The formal expression of δ can be rewritten into the system form as follows

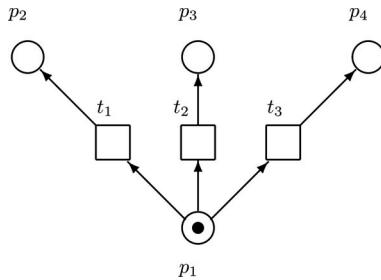


Figure 1. The PNs-based model of the fork of routes.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k, \quad k = 0, \dots, N; \quad \mathbf{B} = \mathbf{G}^T - \mathbf{F}; \quad \mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k \quad (1)$$

where k is the discrete step of the dynamics development; $\mathbf{x}_k = (\sigma_{p_1}^k, \dots, \sigma_{p_n}^k)^T$ is the n -dimensional state vector; $\sigma_{p_i}^k \in \{0, 1, \dots, c_{p_i}\}$, $i = 1, \dots, n$ express the states of atomic activities by 0 (passivity) or by $0 < \sigma_{p_i}^k \leq c_{p_i}$ (activity); c_{p_i} is the capacity of p_i ; $\mathbf{u}_k = (\gamma_{t_1}^k, \dots, \gamma_{t_m}^k)^T$ is the m -dimensional control vector; its components $\gamma_{t_j}^k \in \{0, 1\}$, $j = 1, \dots, m$ represent occurring of elementary discrete events (e.g. starting or ending the activities, failures, etc.) by 1 (presence of the corresponding discrete event) or by 0 (absence of the event); \mathbf{B} , \mathbf{F} , \mathbf{G} are matrices of integers; $\mathbf{F} = \{f_{ij}\}_{n \times m}$, $f_{ij} \in \{0, \mu_{f_{ij}}\}$, expresses the causal relations among the states (as causes) and the discrete events occurring during the DES (discrete-event systems) operation (as consequences) by 0 (nonexistence of the relation) or by $\mu_{f_{ij}} > 0$ (existence and multiplicity of the relation); $\mathbf{G} = \{g_{ij}\}_{m \times n}$, $g_{ij} \in \{0, \mu_{g_{ij}}\}$, expresses analogically the causal relations among the discrete events (as causes) and the DES states (as consequences); $\mathbf{B} = \{b_{ij}\}_{n \times m}$ is given according to Eq. (1), i.e. $b_{ij} = g_{ji} - f_{ij}$; $(\cdot)^T$ symbolizes the matrix or vector transposition. Just such an exact mathematical expression of P/T PN, in contrast to high-level PN, yields the possibility to deal with the PN models in analytical terms.

In the next section an algorithm for finding least-cost transition firing sequences efficiently will be developed. The previous partial results [7,8] will be utilized on this way.

3. Finding Feasible Paths and Transitions Sequences

Let us model the routes (e.g. inside a town or among towns) by the P/T PN arcs containing the P/T PN transitions while the real places where two or more routes intersect, join or fork will be modelled by the P/T PN places. In Figure 1 we can see the situation when p_2 can be reached from p_1 by the route represented by the arc with t_1 . Analogically, p_3 can be reached from p_1 by the arc with t_2 and p_4 can be reached from p_1 by the arc with t_3 . The token in place p_1 models a vehicle at an crossroads and the set of transitions $T = \{t_1, t_2, t_3\}$ denotes the possible behavior of the vehicle as: t_1 means turn left; t_2 means straight; t_3 means turn right. Consider, \mathbf{x}_0 , \mathbf{x}_N being, respectively, the initial state vector, and the desired terminal state vector expressing the initial and terminal locations of the vehicle. We have to solve two principle problems: (i) to find the feasible paths from \mathbf{x}_0 to \mathbf{x}_N ; (ii) to find either shortest one(s) (in case of normal situation in the transport) or the least-cost alternative(s) (in case of congestion in the transport by the shortest paths).

The P/T PN in general (and state machines (SM) as well) can be understood to be bipartite directed graphs (BDG). SM represent a simple kind of P/T PN with the specific structure - each SM transition has only single input place and only single output place. Let $S = \{P, T\}$ is the set of the BDG nodes where P is the set of the PN places and T is the set of the PN transitions. Let Δ is the set $S \times S$ of BDG edges. Their occurrence can be expressed by the $((n+m) \times (n+m))$ adjacency matrix

$$\Delta = \begin{pmatrix} \mathbf{0}_{n \times n} & \mathbf{G}^T \\ \mathbf{F}^T & \mathbf{0}_{m \times m} \end{pmatrix} \quad (2)$$

where $\mathbf{0}_{i \times j}$ in general is the $(i \times j)$ zero matrix; \mathbf{G}, \mathbf{F} are the same incidence matrices like in Eq. (1). In [7,8] the straight-lined reachability tree (developed from the initial state \mathbf{x}_0 towards a prescribed terminal state \mathbf{x}_N) and the backtracking reachability tree (developed from the prescribed terminal state \mathbf{x}_N towards \mathbf{x}_0 , however, oriented from \mathbf{x}_0 towards the terminal state) were found. Then, after a specific kind of intersection of both trees the feasible trajectories from \mathbf{x}_0 to \mathbf{x}_N are found. Analogically, let us use such an approach and apply it to the matrix in Eq. (2). Use the initial vector \mathbf{s}_0 and the terminal vector \mathbf{s}_{2N} . Thus, the straight-lined development (finding the straight-lined reachability tree) is the following

$$\{\mathbf{s}_{k+1}\} = \Delta \cdot \{\mathbf{s}_k\}, k = 0, 1, \dots, 2N-1 \quad (3)$$

$$\mathbf{s}_0 = (\mathbf{x}_0^T, \mathbf{0}_m^T)^T$$

$$\{\mathbf{s}_1\} = (\mathbf{0}_n^T, \{\mathbf{u}_0\}^T)^T; \{\mathbf{s}_2\} = (\{\mathbf{x}_1\}^T, \mathbf{0}_m^T)^T; \{\mathbf{s}_3\} = (\mathbf{0}_n^T, \{\mathbf{u}_1\}^T)^T$$

...

$$\{\mathbf{s}_{2N-1}\} = (\mathbf{0}_n^T, \{\mathbf{u}_{N-1}\}^T)^T; \{\mathbf{s}_{2N}\} = (\{\mathbf{x}_N\}^T, \mathbf{0}_m^T)^T \quad (4)$$

where $\mathbf{0}_j$ in general is the j -dimensional zero column vector. In general, $\{\mathbf{s}_j\}$ is an aggregate all of the states that are reachable from the previous states $\{\mathbf{s}_{j-1}\}$. The straight-lined reachability tree arising from the initial vector \mathbf{s}_0 and directed towards the terminal state \mathbf{s}_{2N} can be recorded by the matrix

$$\mathbf{M}_1 = (\mathbf{s}_0, {}^1\{\mathbf{s}_1\}, \dots, {}^1\{\mathbf{s}_{2N-1}\}, {}^1\{\mathbf{s}_{2N}\}) \quad (5)$$

The backtracking development (finding the backtracking reachability tree) is as follows

$$\{\mathbf{s}_{2N-k-1}\} = \Delta^T \cdot \{\mathbf{s}_{2N-k}\}, k = 0, 1, \dots, 2N-1 \quad (6)$$

$$\mathbf{s}_{2N} = (\mathbf{x}_N^T, \mathbf{0}_m^T)^T; \{\mathbf{s}_{2N-1}\} = (\mathbf{0}_n^T, \{\mathbf{u}_{N-1}\}^T)^T$$

$$\{\mathbf{s}_{2N-2}\} = (\{\mathbf{x}_{N-1}\}^T, \mathbf{0}_m^T)^T; \{\mathbf{s}_{2N-3}\} = (\mathbf{0}_n^T, \{\mathbf{u}_{N-2}\}^T)^T$$

...

$$\{\mathbf{s}_1\} = (\mathbf{0}_n^T, \{\mathbf{u}_0\}^T)^T; \{\mathbf{s}_0\} = (\{\mathbf{x}_0\}^T, \mathbf{0}_m^T)^T \quad (7)$$

The backtracking reachability tree is recorded by the matrix

$$\mathbf{M}_2 = ({}^2\{\mathbf{s}_0\}, {}^2\{\mathbf{s}_1\}, \dots, {}^2\{\mathbf{s}_{2N-1}\}, \mathbf{s}_{2N}) \quad (8)$$

The intersection of the trees is made as follows

$$\mathbf{M} = \mathbf{M}_1 \cap \mathbf{M}_2 \quad (9)$$

where the matrices \mathbf{M}_1 , \mathbf{M}_2 represent, respectively, the straight-lined tree and the backtracking one. The special intersection both of the trees is performed by means of the column-to-column intersection both of the matrices.

After the intersection according to Eq. (9) we have

$$\mathbf{M} = (\mathbf{s}_0, \{\mathbf{s}_1\}, \dots, \{\mathbf{s}_{2N-1}\}, \mathbf{s}_{2N}) \quad (10)$$

where $\{\mathbf{s}_i\} = \min(1\{\mathbf{s}_i\}, 2\{\mathbf{s}_i\})$, $i = 0, 2N$ with $1\{\mathbf{s}_0\} = \mathbf{s}_0$, $2\{\mathbf{s}_{2N}\} = \mathbf{s}_{2N}$.

Information about the state vectors is stored in the even columns of \mathbf{M} while information about the control vectors is stored in its odd columns.

Because of the special block form of both the matrix Δ and the vectors \mathbf{s}_k , $k = 0, 2N$ we can simultaneously use (alternate step-by-step) two procedures with dimensionality n, m , respectively.

The step-by-step alternation is used to improve the previous procedure. Let us use the alternative straight-lined procedure as follows, where instead storing the vectors into \mathbf{M}_1 , \mathbf{M}_2 the step-by-step alternation of storing the vectors into ${}^j\mathbf{X}$, ${}^j\mathbf{U}$, $j = 1, 2$ is used. Such an approach yields two advantageous: (i) the usage of zero blocks of the matrix Δ is eliminated; (ii) both the state trajectories (paths) and the control trajectories (sequences of the transitions) are found.

$$\begin{aligned} {}^1\mathbf{X} &= (\mathbf{x}_0, \emptyset_{n \times N}) \\ {}^1\{\mathbf{u}_0\} &= \mathbf{F}^T \cdot \mathbf{x}_0; {}^1\mathbf{U} = ({}^1\{\mathbf{u}_0\}, \emptyset_{m \times (N-1)}) \\ {}^1\{\mathbf{x}_1\} &= \mathbf{G}^T \cdot {}^1\{\mathbf{u}_0\}; {}^1\mathbf{X} = (\mathbf{x}_0, {}^1\{\mathbf{x}_1\}, \emptyset_{n \times (N-1)}) \\ {}^1\{\mathbf{u}_1\} &= \mathbf{F}^T \cdot {}^1\mathbf{x}_1; {}^1\mathbf{U} = ({}^1\{\mathbf{u}_0\}, {}^1\{\mathbf{u}_1\}, \emptyset_{m \times (N-2)}) \\ {}^1\{\mathbf{x}_2\} &= \mathbf{G}^T \cdot {}^1\{\mathbf{u}_1\}; {}^1\mathbf{X} = (\mathbf{x}_0, {}^1\{\mathbf{x}_1\}, {}^1\{\mathbf{x}_2\}, \emptyset_{n \times (N-2)}) \\ &\dots \quad \dots \quad \dots \\ {}^1\{\mathbf{u}_{N-1}\} &= \mathbf{F}^T \cdot {}^1\mathbf{x}_{N-1}; {}^1\mathbf{U} = ({}^1\{\mathbf{u}_0\}, {}^1\{\mathbf{u}_1\}, \dots, {}^1\{\mathbf{u}_{N-1}\}) \\ {}^1\{\mathbf{x}_N\} &= \mathbf{G}^T \cdot {}^1\{\mathbf{u}_{N-1}\}; {}^1\mathbf{X} = (\mathbf{x}_0, {}^1\{\mathbf{x}_1\}, {}^1\{\mathbf{x}_2\}, \dots, {}^1\{\mathbf{x}_N\}) \end{aligned}$$

where ${}^1\mathbf{U}$ is $(m \times N)$ matrix and ${}^1\mathbf{X}$ is $(n \times (N+1))$ matrix. The left upper index ${}^1(\cdot)$ points out performing the straight-lined procedure. The backtracking procedure run as follows

$$\begin{aligned} {}^2\mathbf{X} &= (\emptyset_{n \times N}, \mathbf{x}_N) \\ {}^2\{\mathbf{u}_{N-1}\} &= \mathbf{G} \cdot \mathbf{x}_N; {}^2\mathbf{U} = (\emptyset_{m \times (N-1)}, {}^2\{\mathbf{u}_{N-1}\}) \\ {}^2\{\mathbf{x}_{N-1}\} &= \mathbf{F} \cdot {}^2\{\mathbf{u}_{N-1}\}; {}^2\mathbf{X} = (\emptyset_{n \times (N-1)}, {}^2\{\mathbf{x}_{N-1}\}, \mathbf{x}_N) \\ {}^2\{\mathbf{u}_{N-2}\} &= \mathbf{G} \cdot {}^2\{\mathbf{x}_{N-1}\}; {}^2\mathbf{U} = (\emptyset_{m \times (N-1)}, {}^2\{\mathbf{u}_{N-2}\}, {}^2\{\mathbf{u}_{N-1}\}) \\ {}^2\{\mathbf{x}_{N-2}\} &= \mathbf{F} \cdot {}^2\{\mathbf{u}_{N-1}\}; {}^2\mathbf{X} = (\emptyset_{n \times (N-2)}, {}^2\{\mathbf{x}_{N-2}\}, {}^2\{\mathbf{x}_{N-1}\}, \mathbf{x}_N) \\ &\dots \quad \dots \quad \dots \end{aligned}$$

$$\begin{aligned} {}^2\{\mathbf{u}_1\} &= \mathbf{G}.{}^2\{\mathbf{x}_2\}; {}^2\mathbf{U} = (\emptyset_{n \times 1}, {}^2\{\mathbf{u}_1\}, \dots, {}^2\{\mathbf{u}_{N-1}\}) \\ {}^2\{\mathbf{x}_1\} &= \mathbf{F}.{}^2\{\mathbf{u}_1\}; {}^2\mathbf{X} = (\emptyset_{n \times 1}, {}^2\{\mathbf{x}_1\}, {}^2\{\mathbf{x}_2\}, \dots, \mathbf{x}_N) \\ {}^2\{\mathbf{u}_0\} &= \mathbf{G}.{}^2\{\mathbf{x}_1\}; {}^2\mathbf{U} = ({}^2\{\mathbf{u}_0\}, {}^2\{\mathbf{u}_1\}, \dots, {}^2\{\mathbf{u}_{N-1}\}) \\ {}^2\{\mathbf{x}_0\} &= \mathbf{F}.{}^2\{\mathbf{x}_0\}; {}^2\mathbf{X} = ({}^2\{\mathbf{x}_0\}, {}^2\{\mathbf{x}_1\}, {}^2\{\mathbf{x}_2\}, \dots, \mathbf{x}_N) \end{aligned}$$

where ${}^2\mathbf{U}$ is $(m \times N)$ matrix and ${}^2\mathbf{X}$ is $(n \times (N+1))$ matrix. The left upper index ${}^2(.)$ points out performing the backtracking procedure.

The final phase of the control problem solving is the special intersection described above. In such a way we have both the system state trajectories and corresponding control strategies

$$\mathbf{X} = {}^1\mathbf{X} \cap {}^2\mathbf{X}; \quad \mathbf{X} = (\mathbf{x}_0, \{\mathbf{x}_1\}, \dots, \{\mathbf{x}_{N-1}\}, \mathbf{x}_N) \quad (11)$$

$$\mathbf{U} = {}^1\mathbf{U} \cap {}^2\mathbf{U}; \quad \mathbf{U} = (\{\mathbf{u}_0\}, \{\mathbf{u}_1\}, \dots, \{\mathbf{u}_{N-1}\}) \quad (12)$$

Using zero blocks in Eq. (2) was eliminated. Thus, we have \mathbf{X} where the *state trajectories* are comprehended and \mathbf{U} where the *control trajectories* are comprehended.

3.1. Illustrative Example

Consider a real situation [4] displayed up in Figure 2, where the vehicle is ready to move from the location A to B . Assume that the distance of each road section is given by $d_{12} = 40\text{m}$, $d_{24} = 30\text{m}$, $d_{13} = 30\text{m}$, $d_{34} = 20\text{m}$, $d_{45} = 50\text{m}$, $d_{65} = 30\text{m}$, $d_{76} = 30\text{m}$, $d_{73} = 40\text{m}$, $d_{37} = 40\text{m}$, $d_{87} = 40\text{m}$, $d_{18} = 30\text{m}$, where d_{ij} is the distance from intersection point i to j . These distances can be incorporated into Petri net models as transition costs in case when only the shortest path(s) is(are) found. However, the traffic flow situation of each route is different so that the travel time needed is different. In case when data about the average vehicle speed of each road section (because of a kind of congestion) are at disposal from AGTPM system, we can used such information to compute the average travel time (cost) given to the fixed distance of each road section. Assume that the average vehicle speed of each road section is estimated by an AGTPM system as $v_{12} = 10\text{m/s}$, $v_{24} = 5\text{m/s}$, $v_{13} = 5\text{m/s}$, $v_{34} = 5\text{m/s}$, $v_{45} = 10\text{m/s}$, $v_{65} = 10\text{m/s}$, $v_{76} = 3.33\text{m/s}$, $v_{73} = 10\text{m/s}$, $v_{37} = 5\text{m/s}$, $v_{87} = 10\text{m/s}$, $v_{18} = 10\text{m/s}$ (where m/s stands for meters per second), the estimated travel time τ_{ij} can be computed as the ratio of the distance to the average vehicle speed and assigned to the corresponding transition as $(\tau_{12} = 4\text{s}) \rightarrow t_1$, $(\tau_{24} = 6\text{s}) \rightarrow t_2$, $(\tau_{13} = 6\text{s}) \rightarrow t_3$, $(\tau_{34} = 4\text{s}) \rightarrow t_4$, $(\tau_{45} = 5\text{s}) \rightarrow t_5$, $(\tau_{56} = 3\text{s}) \rightarrow t_6$, $(\tau_{76} = 9\text{s}) \rightarrow t_7$, $(\tau_{73} = 4\text{s}) \rightarrow t_8$, $(\tau_{37} = 8\text{s}) \rightarrow t_9$, $(\tau_{87} = 4\text{s}) \rightarrow t_{10}$, $(\tau_{18} = 3\text{s}) \rightarrow t_{11}$ (where s stands for seconds). These estimated travel times can be incorporated into Petri net models as transition costs when the shortest time (or better, least-time) cost route(s) are found.

The PN model of the transportation system is in Figure 2 down on the left. The corresponding reachability graph in Figure 2 down on the right shows different travel routes leading to the destination (from \mathbf{x}_0 to \mathbf{x}_7). The PN has 8 places (i.e. intersections) and 11 transitions. The PN initial marking is given by $\mathbf{x}_0 = (10000000)^T$ (stored the 1-st column of the below introduced matrix \mathbf{X}_{reach}) and the final marking is $\mathbf{x}_7 = (00000100)^T$ (stored in the 8-th column of \mathbf{X}_{reach}). In order to go from the initial marking to the final marking (to drive the vehicle from point A to point B), there are different transition firing sequences (indicating travel routes) of different length (expressing the number of

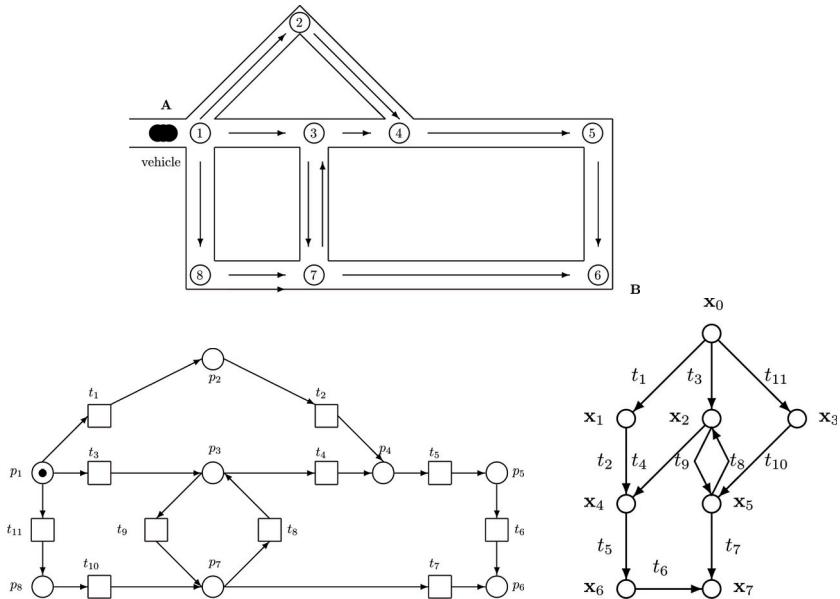


Figure 2. A real scenario of the transportation (up), its P/T PN model (down on the left) and the reachability graph - it yields the feasible routes from the state x_0 (representing the location A) to the state x_7 (representing the location B) (down on the right).

segments to be passed) - e.g. $\{t_1, t_2, t_5, t_6\}$, $\{t_3, t_4, t_5, t_6\}$, $\{t_3, t_9, t_7\}$, $\{t_{11}, t_{10}, t_7\}$, etc. The incidence matrices of the PN model and the initial state are the following

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}; \mathbf{G}^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The symbolic adjacency matrix \mathbf{A} of the reachability graph (its entries represent the indices of transitions) and the matrix \mathbf{X}_{reach} containing the feasible states (x_0 and all states reachable from x_0) as its columns are the following

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 3 & 11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Table 1. The computational results

Number of segments	Firing Sequence	Path Length [m]	Time [s]	Least-Cost Solution
3	$t_3t_9t_7$ $t_{11}t_{10}t_7$	100 100	23 16	— 16s
4	$t_1t_2t_5t_6$	150	18	—
	$t_3t_4t_5t_6$	130	18	130m
5	$t_3t_9t_8t_9t_7$	180	35	—
	$t_{11}t_{10}t_8t_9t_7$	180	28	28s
6	$t_3t_9t_8t_4t_5t_6$	210	30	—
	$t_{11}t_{10}t_8t_4t_5t_6$	210	23	23s
7	$t_3t_9t_8t_9t_8t_9t_7$	260	47	—
	$t_{11}t_{10}t_8t_9t_8t_9t_7$	260	40	40s

Having \mathbf{U} , multiplying this matrix from the left by the diagonal matrix \mathbf{D} with d_{ij} in its diagonal the *control trajectories* are calibrated by the distances. Multiplying \mathbf{U} from the left by the diagonal matrix \mathbf{T}_τ with τ_{ij} in its diagonal the *control trajectories* are calibrated by the times. The number of columns of \mathbf{U} is equal to the number μ of SM transitions to be passed in order to reach x_7 from x_0 . In our case $\mu \in \{3, 4, 5, 6, 7\}$. For instance, in case of $\mu = 3$ we have

$$\mathbf{U} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}; \mathbf{D} \cdot \mathbf{U} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 30 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 30 \\ 0 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 40 & 0 \\ 30 & 0 & 0 \end{pmatrix}; \mathbf{T}_\tau \cdot \mathbf{U} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 9 \\ 0 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 4 & 0 \\ 3 & 0 & 0 \end{pmatrix}$$

3.1.1. Results of Computations

The shortest transitions firing sequence is 3 while the reasonable longest one is 7 (it contains also the loop t_8, t_9) - see Table 1. As we can see from Table 1 the least-cost solution is in its first row - the framed one.

4. Agent Based Approach

Considering the scenario up in Figure 2 to be the agent A1 we can define another (adjacent) scenario (e.g. movement from the point B to a point C) being another agent A2. The structural matrices of both *cooperating* agents are

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_{c_1} \\ \mathbf{F}_{d_1} & \mathbf{F}_2 \end{pmatrix}; \mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{G}_{d_1} \\ \mathbf{G}_{c_1} & \mathbf{G}_2 \end{pmatrix} \quad (13)$$

where the pairs \mathbf{F}_{c_1} , \mathbf{G}_{c_1} and \mathbf{F}_{d_1} , \mathbf{G}_{d_1} represent, respectively, interconnections between A1 and A2 throughout the transitions and throughout the places. When the point B need not be passed, we can apply the previous algorithm on \mathbf{F} , \mathbf{G} given in Eq. (13) in order to find least-cost solution for movement from the point A to the point C. Of course, another adjacent agents A3, A4, etc., can be successively added (e.g. by virtue of the plan of city, map of country, etc.). When only movement from the point B to point C is necessary, it is sufficient to apply the previous algorithm on the matrices \mathbf{F}_2 , \mathbf{G}_2 describing the A2.

To avoid congestion the amount of information available within cooperative system is very useful. The advantages of the agent approach consist in the fact that the approach (i) can yield information about all segments of the transport in a chosen global area (e.g. the whole city or a part of country) simultaneously; (ii) subsequently, solving the problems in two or more adjacent local areas (directly influencing each other) is more flexible and more effective. Communication among agents is one of the important properties of cooperative systems. In comparison with other approaches (agent and non-agent based) the agent approach presented here is more advantageous especially from two computational points of view. Namely, the approach is linear and described in analytical terms. Consequently, the process of simulation is easy.

5. Conclusion

The algorithm for finding the least-cost paths from a given starting point of a vehicle to the prescribed terminal point was presented in this paper. It is based on P/T PN model, more precisely on SM model (SM are special kinds of P/T PN where each transition has only one input place and only one output place). The SM places represent the points (e.g. crossroads) to be passed in order to reach the terminal point while the weighted transitions represent the segments of the paths between two points to be passed. The algorithm based on the intersection of the straight-lined reachability tree and the backtracking one yields both the *state trajectories* consisting of the sequence of the SM places to be passed and the *control trajectories* consisting of the sequence of the SM transitions. The SM transitions can be weighted by the lengths of the corresponding segments. However, when some congestions occur on the shortest segments, alternative segments with actual least-cost throughput (in the form of a time necessary for passing the segments computed from data about the actual speed obtained from AGTTPM) can be found - it means that the vehicle can reach the terminal point sooner than along the shortest path.

Acknowledgement

The theoretical research was partially supported by the Slovak Grant Agency for Science (VEGA) under the current grants #2/0075/09, # 2/0039/13. The theoretical results were applied at solving the practical project, videlicet: This contribution is the result of the project implementation: Technology research for the management of business processes in heterogeneous distributed systems in real time with the support of multimodal communication, code ITMS: 26240220064, supported by Operational Programme Research & Development funded by the ERDF. The author thanks both institutions for the support of his research.

References

- [1] W. van Hemert, *Can Cooperative Systems Contribute to Reducing Traffic Congestions?*, Department of Information and Computing Sciences, Utrecht University, The Netherlands, Logica, 2010.
- [2] Y. Liu, Y. Dai and Z. Dai, Real-time adaptive gray threshold measurement in extracting traffic parameters, Proceedings of the 3rd International Symposium on Computational Intelligence and Industrial Applications, Dali, Yunnan Province, China, pp. 255–265, 2008.
- [3] L. Li, Y. Ru and C.N. Hadjicostis, Least-cost transition firing sequence estimation in labeled Petri nets, Proceedings of the 45th IEEE International Conference on Decision and Control, San Diego, CA, USA, pp. 416–421, 2006.
- [4] Y. Qu, L. Li, Y. Liu, Y. Chen and Y. Dai, Travel routes estimation in transportation systems modeled by Petri nets, Proceedings of the 2010 IEEE International Conference on Vehicular Electronics and Safety, Qing Dao, China, pp. 73–77, 2010.
- [5] C. Tolba, D. Lefebvre, P. Thomas and A. El Moudni, Continuous and timed Petri nets for the macroscopic and microscopic traffic flow modelling, *Simulation Modelling Practice and Theory* **13** (2005), 407–436.
- [6] T. Murata, Petri nets: properties, analysis and applications, *Proceedings of the IEEE* **77** (1989), 541–580.
- [7] F. Čapkovič, Automatic control synthesis for agents and their cooperation in MAS, *Computing and Informatics* **29** (2010), 1045–1071.
- [8] F. Čapkovič, Agent based approach to modelling ATM network, Proceedings of the 2012 IEEE 6th International Conference on Intelligent Systems, Sofia, Bulgaria, IEEE Press, Piscataway, NJ, USA, pp. 102–107, 2012.

On Scalable, Event-Oriented Control for Lighting Systems

Igor WOJNICKI ^{a,1}, Leszek KOTULSKI ^a and Sebastian ERNST ^a

^aAGH University of Science and Technology, Department of Applied Computer Science,
Kraków, Poland

Abstract. This paper proposes a scalable, multi-agent architecture for control of modern outdoor lighting systems. Most contemporary lighting systems utilize a static control structure, which is based on simple criteria (e.g. date, time of day, weather forecast) and operate on few (two or three) lighting modes of luminaires. Thus, centralized management is sufficient for such systems. Modern lighting control systems take dynamic and fine-grained (often local) conditions into account and operate on more sophisticated equipment, characterized by flexible lighting levels and geometries. These factors cause scalability problems, which may render the system unable to react to incoming events in time. The proposed solution introduces a hierarchy of agents, which allow for distributed control and supervision. Moreover a graph-based model is introduced as a formal representation of the agent's knowledge, which allows it to be processed in parallel by distributed agents.

Keywords. lighting control, intelligent control, agents, graph structure, outdoor lighting

1. Motivation

Contemporary lighting systems are characterized by a static control structure with light points operating in a few possible modes (usually three: *on*, *saving* and *off*). Sometimes, static daily performance profiles (being sequences of a few subsequent power levels) are defined. Another property of such systems is centralized management of an entire, large area, which could pose difficulties if prompt responses need to be provided to events occurring in a local environment. Let us notice that events occurring at the same time in various locations should be handled in a parallel way. Thus, a more flexible software solution should be applied to control modern lighting systems with the above characteristics. Control should address functional properties of a lighting system (in particular, compliance with lighting standards) as well as optimization (minimization) of energy usage. In this paper, we suggest to solve this problem with the help of a hierarchical agent structure² (see Section 4). We will use a graph notation as a formalism which contains the global knowledge about the entire system as well as local knowledge of an individual

¹Corresponding Author: Igor Wojnicki, AGH University of Science and Technology, Department of Applied Computer Science, Al. A. Mickiewicza 30, 30-059 Kraków, Poland; E-mail: wojnicki@agh.edu.pl.

²Work has been co-financed by the European Union, Human Capital Operational Programme, SPIN project no. 502.120.2066/C96 and co-financed by the Research Fund no. 11.11 120.859.

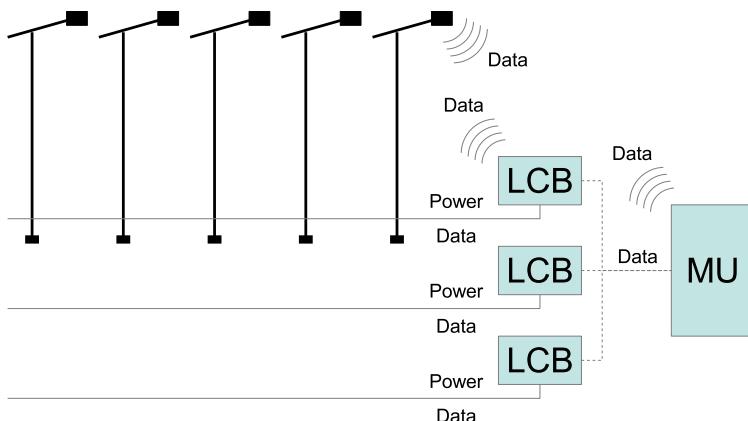


Figure 1. Contemporary street lighting.

agent. Parallel cooperation among agents forces us to solve the problems related to their communication and synchronization.

2. Contemporary Hierarchical Lighting Model

Contemporary street lighting systems form hierarchical or semi-hierarchical structures (see Figure 1). Luminaries within lamps are directly connected to power lines. Enabling or disabling power turns light points on or off. The power lines are managed by lighting control boxes (LCBs). A single lighting control box handles a given area, set of luminaries which often spreads over a street or a block.

A basic structure assumes that enabling or disabling power to luminaries are the only means of control. However, the power lines might be redundant, providing dimming, which in turn enables two or more luminosity levels. Considering advanced lighting systems, a control box can communicate with selected luminaries, directing them to switch to a given power output using the Power Line Communication (PLC) protocol or wireless technologies. The communication can also be bi-directional, which allows the control box to gather data from the luminaries regarding their working parameters.

The control box decides about turning luminaries to desired power output levels based on internal, software-defined logic. In most cases, the decision is based on the current time and date; however, it can be freely programmed.

Optionally, especially for intelligent lighting systems, the control boxes can be connected with a central managing unit (MU). The unit provides control box management and control logic updates.

Contemporary street lighting systems, while still flexible, thanks to the hierarchical approach, are often hard to manage or update. While having a central managing unit helps with software or control logic updates, it does not solve all issues. For example, adding or removing luminaries within an existing infrastructure poses a challenge when assigning or reassigning them to control boxes in order to support consistent control over spatially distributed light points.

3. Control

Control of lighting systems can be classified as *passive* or *active*, depending on the conditions used in the decision process. *Passive* control is based on conditions which are predictable – which can be determined without up-to-date data, thus allowing control routines to be determined *a priori* for the entire time of system operation. On the other hand, *active* control relies on current conditions, usually gathered from real-time sensors. As these conditions cannot be fully predicted, the system needs to *react* to them in a given time frame.

Conditions which can be used as input for *passive* lighting control systems include the following:

- **location** and **current time and date** are commonly used to estimate sunset and sunrise times, which imply the operating hours,
- **weather forecast**, as opposed to current weather data provided by sensors, can be used to predict times when ambient light drops below acceptable levels, creating the need for operation of luminaries.

Introduction of *active* control makes it possible to use more fine-grained conditions for control, including:

- **motion** and **presence** detection, which can be used to increase lighting intensity where pedestrians or vehicles are present,
- **light sensors**, which can be used for more precise control of luminaries:
 - * *ambient light sensors* make it possible to calculate the precise intensity needed to compensate for prevailing lighting conditions,
 - * *reflected light sensors* allow for closed-loop control of lighting intensity,
- **current weather sensors** can be used to validate forecasted conditions and react to unpredicted changes.

While *passive* control scales well and can be performed in a centralized manner, *active* control can result in performance issues (which can cause control instructions to be generated too late), especially in the event of an event flood. Moreover, a significant part of recorded events only influence a certain region and may not impact the entire system. Therefore, such decisions should be made locally, improving the processing load distribution and the performance of the system.

4. Multi-agent System Architecture

Assumptions concerning the environment model and system goals imply the high complexity of related computations, making the “naïve” models which assume a simple geometric representation of the entire urban space ineffective.

To provide an insight into the scale of the problem, we shall use an example case of a LED (light emitting diode) fixture. It may be assumed that the considered fixture can operate at 50 possible power levels. It consists of multiple micro light sources (diodes) coupled with individual optics being lenses or mirrors. Photometric parameters of the micro light sources can be individually configured or controlled. It results in a possibility

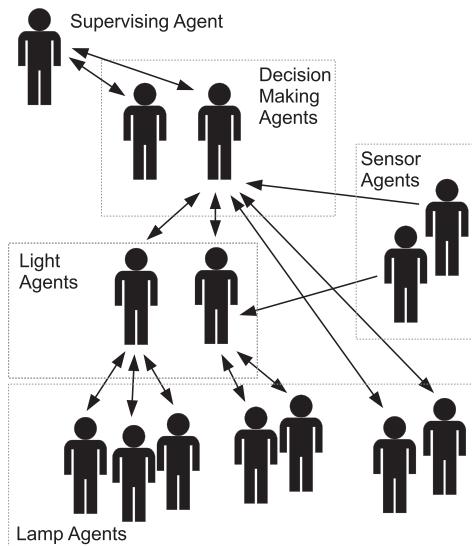


Figure 2. Architecture of Proposed Multi-agent System

of having multiple, different light distributions from the same, single fixture. Assuming that 256 light distributions are possible, that amounts to 12,800 possible fixture states, not counting other degrees of freedom such as the fixture's azimuth, inclination, etc.

This simple case shows that the aforementioned simple geometric representations are insufficient to bear computational complexity. Therefore, we propose to replace them with graph representations. The proposal is based on the assumption that we will be able to split the graph representing the system into a set of subgraphs. Each subgraph will represent sufficient knowledge for the corresponding agent to make an appropriate decision.

A multi-agent approach makes it possible to execute control tasks regardless of the number of light points. However, this generates the problem of explicit synchronization of concurrently working agents. Avoiding time-dependent errors for explicit synchronization requires significant effort and is a complex issue. For this reason, an implicit synchronization method based on separation of data modifiable by agents (see Section 5) is proposed.

The proposed Scalable, Event-Oriented Control (SEOC) utilizes the following types of FIPA-compliant agents, assigned either to physical components (lamps, sensors) or to abstract entities as software modules (Figure 2).

- **Sensor Agent (NA)** – collects data captured by a sensor unit and relays them to *light agents* or a Decision Making Agent (see below). It runs on the sensor device (within a fixture, lamp post, road infrastructure, weather station), LCB or MU (see Figure 1), depending on its type.
- **Lamp Agent (LA)** – located over the lamp driver layer. This agent keeps direct control, via a lamp driver, over lamp behavior (dimming, switching on/off) and intermediates between a Light Agent (or Decision Making Agent) and a driver. It transmits all exploitation/diagnostic data concerning the state of the device. It runs on the lamp driver, or on LCB.

- **Decision-Making Agent (DMA)** – main control agent. It is responsible for collecting sensory/diagnostic data, lighting control in the *empty mode* (e.g. when no objects are present in the scene), creating *light agents* when objects appears in a controlled area. It runs on LCB or MU.
- **Light Agent (IA)** – manages a set of individual luminaries via a lamp agent. If some object(s) appears in a scene, a DMA receives information from sensor(s) and determines the set of lamps involved in illuminating the corresponding area. Next, a light agent is created by a DMA and ascribed to that set. The control tasks are delegated to this light agent together with a relevant subgraph representing that area and the entities being controlled (in particular luminaries). It runs on LCB.
- **Supervising Agent (SA)** – manages *decision-making agents*, ensuring proper spawning and supervision, including software and behavior updates. It constitutes a supervision tree which is a proven and reliable approach known from high availability telecommunication systems such as these based on Erlang/OTP (Open Telecom Platform)[1]. It runs on MU.

Agents in the proposed multi-agent approach implement different levels of sophistication depending on their type. Three levels are considered here: *low*, *medium* and *high*.

The *low* level regards Sensor and Lamp Agents. They represent mainly reactive behavior; their main goal is to relay data ensuring two-way communication and bare data validation, including assessment of possible device failures (due to garbled data stream). Such agents are also partially autonomous, capable of taking actions if communication with higher level agents is compromised.

The *medium* level regards Decision Making Agents and the Supervising Agent. Their main goals are handling passive control and providing supervision for other agents: Light Agents in case of a Decision Making Agent, or Decision Making Agents in case of the Supervising Agent. They use graph transformations and rules to identify proper subgraphs to be assigned to other agents.

The *high* level regards Light Agents. They handle dynamically-changing environment and provide active control. They use graph transformations, triggered by sensor data patterns and a rule-based approach.

5. Graph-based Representation

To formally model spatial distribution and properties of physical objects under consideration, a General Environment Model (GEM) is proposed. It consists of architectural space, light points, light control boxes and sensor devices. The architectural space represents an urban environment: buildings, roads, sidewalks, lawns, park areas, etc. A light point (a luminary) is characterized by a predefined set of properties: coordinates, pole height, fixture type (inclination, power, light distribution), lamp fixture overhang, etc. Similarly, sensors have their properties and spatial distribution defined as well.

Graph and hypergraph models are useful representations in the lighting design tasks[2]. Let us consider an example environment being a square with adjacent buildings. Its plan is given in Figure 3a while corresponding hypergraph representing it is given in Figure 3b. The example is simplified. It considers: two facades (B1, B2), one plain, being the square pavement, (B1), two lamps (L3, L4), and six sensors (N1, … N6). Mind that

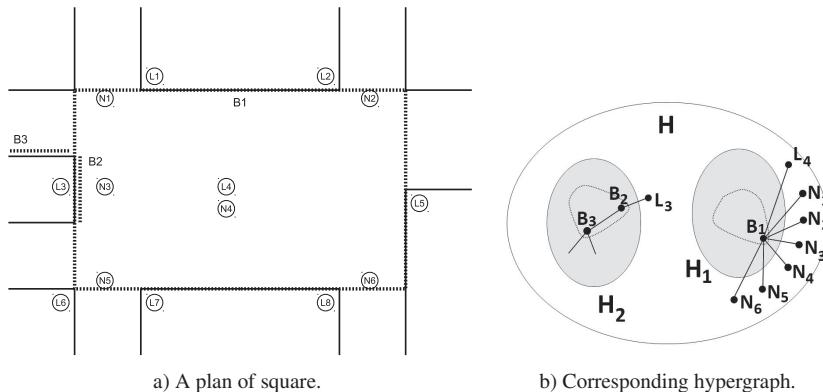


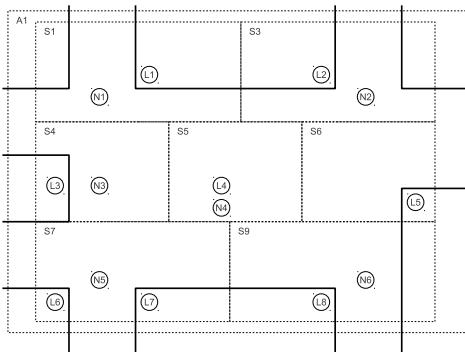
Figure 3. A plan of square with buildings and corresponding hypergraph showing an example of the General Environment Model. B – simple architectural feature, H – complex architectural feature, L – lamp, N – sensor.

varying topologies and functions can be considered this way: streets, squares, public areas, etc. Additional elements such as motion sensors and lamps are labeled respectively as Ns and Ls.

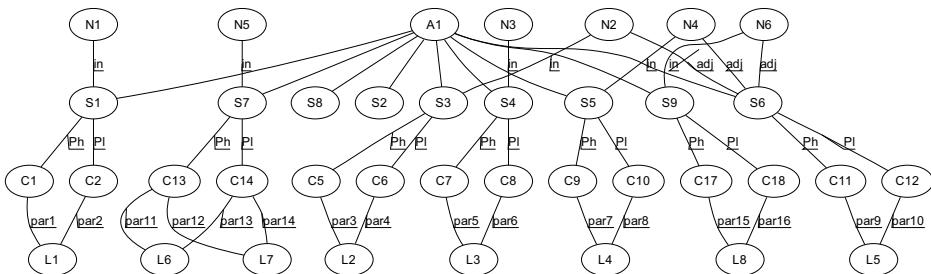
For simplicity, the figure (b) contains only general features of the hypergraph modeling a given architectural space. The area is equipped with a single lamp L4 and sensors: N1, ..., N6. The outer contour represents the main hypergraph, H, describing the entire area. Its two subhypergraphs (indicated as gray shaded ovals), H1 and H2, describe the square and the adjacent building, respectively. Vertices B2, B3 included in H2 represent two adjacent facades of the building. Both vertices lie inside the dotted line, which denotes a hyperedge. Some other H2 edges and hyperedges may be incident with B2 and B3. Additional vertex, L3, representing a lamp mounted on the wall, is neighboring with node B2. The area of the square is represented by node B1. Vertices neighboring with B1 represent related lamps and sensors.

To define relationships among considered areas, luminaries, sensors and actuators, GEM is extended with additional information called Control Availability (CA). An example CA is showed in Figure 4. The proposed formalism is similar to the Control System Influence Graph introduced in [3] which defines direct relationship among sensors and light points; however it has been significantly extended.

CA allows to partition the entire environment into manageable subsections (areas A and segments S), taking into account the incoming data feed from sensors (N). There is a single area A1 presented (Figure 4) subdivided into multiple segments (S1, S3, S4, S5, S6, S7, S9). Sensors (N1, ..., N6) are in relation with segments which in this case are: *in* or *adjacent*. It expressed as labeled edges visible in Figure 4b. It focuses the control process on particular substructures in GEM, depending on data from sensors, instead of considering the entire state space. It also enables DMA to spawn multiple LAs (Light Agents) if there are multiple partitions identified. Having GEM-CA, the control system can match current conditions and a pattern of active sensors and decide to switch to a different profile for a given area or segment. The configuration vertices (C1, C2, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C17, C18) allow to express sets of parameters for particular lamps. A single configuration vertex (C) defines a particular lighting profile for given segment (S). Each edge between C and S is labeled with lighting profile name,



a) A plan of square with buildings subdivided into manageable segments.



b) Corresponding Control Availability.

Figure 4. A plan of square with buildings subdivided into manageable segments and corresponding Control Availability. Vertices: A – area, S – segment, L – lamp, N – sensor, C – configuration. Edges: P – light profile, par – luminary parameters, adj – adjacent to segment (spatial relationship), in – in segment (spatial relationship).

the configuration provides, while each edge between C and L with particular parameters for the given lamp.

As mentioned before, the graph-based formal model is necessary to handle the complexity of lighting control. To make the graph representation a suitable environment for multi-agent system deployment, decomposition of a centralized graph has to be performed. The important issue related to this step is selecting a proper decomposition method. Using an inappropriate approach may significantly increase the complexity of agents' synchronization. This, in turn, influences response times.

For this reason, we selected the representation based on the concept of *graph slashing* [4]. In this approach, some edges are slashed and (*dummy*) vertices are attached to newly created endpoints. Each dummy node exists in two replicas, shared only by two agents. Thus, the coupling among agents, as well as the complexity of their coordination, is significantly reduced. Good performance of a multi-agent system, operating in such environment was proven in [4] by conducting comparative tests, using a MAS deployed on JADE framework [5].

When the feature of graph replication is required then the representation referred to as Replicated Complementary Graphs (RCG) may be used [6]. In this case, however, a inter-agent coupling may decrease the efficiency of computations.

6. Related Research

Most of the intelligent lighting research focuses on smart buildings. It includes both office and home spaces. Well-defined zones (predefined areas) with sensors deployed are used to obtain information about the state of the environment. Taking into consideration user lighting preferences, inhabitant tracking, daylight intensity detection and window blinding control leads to energy savings [7]. The above solution is based on agents and uses a hidden Markov model for inhabitant tracking.

A similar approach is proposed in [3]. It also focuses on in-building operations. Additionally, the control system either sends control commands to actuators or informs the inhabitants that there is a need to switch certain light points, blinds, etc. to reduce power usage – therefore performing an educational role. Detection of outdoor lighting conditions is based on a complex sky model.

There is an experiment presented in [8,9] conducted in an office space aimed to reduce energy consumption. Having given lighting requirements (desired luminosity at certain locations) and multiple light points, the goal is achieved by turning or dimming some lamps. Simulated Annealing is used to conduct the optimization process.

Outdoor lighting optimization is presented in [10]. A highway tunnel lighting case is discussed. The tunnel is equipped with vehicle and luminance detectors. Also, some luminance requirements are given to comply with safety regulations. Applying the proposed control based on data from sensors leads to energy savings.

Energy savings due to application of intelligent control are confirmed in [11]. Both traffic and weather conditions are taken into consideration, leading to 40.9% energy savings.

All presented approaches indicate a common theme: possible energy savings due to intelligent control, which is often confirmed by the experiments.

The main goals of the proposed approach are: energy savings, increased safety and last but not least aesthetics[12]. Utilizing the proposed agent-based architecture enables to flexibly deploy intelligent control routines which provide the above. Comparing with the approaches presented earlier in this section it is characterized by the following features.

- It is topology-driven, tailored to handling complex topologies, complex lighting setups, formally defined with hypergraphs.
- It is well scalable, employing the agent-based approach.
- Outdoor applications (squares, streets, parks) are targeted.
- It introduces lighting profiles employing modes of operation. The system switches to proper lighting profile upon information from sensors.

It needs to be pointed out that the use of graph and hypergraph representations in the presented approach makes it applicable to problems of any structural complexity. Their expressive power is sufficient to describe complete topologies [13,14]. The added value of this method, compared with others [15], is scalability supported by the mentioned formal model and parallel computation methods based on agents. Furthermore, since it is a formal approach certain design process methodologies and analysis could also be applied [16].

7. Summary and Further Research

This paper introduces a multi-agent architecture for control of modern lighting systems, so-called Scalable Event-Oriented Control (SEOC), which is characterized by stability and distributed processing. Centralized management models are sufficient for lighting systems based on simple (static) criteria, which utilize lamps with two or three light modes. Modern lighting systems, which base control upon dynamic conditions (see Section 3) and operate sophisticated luminaires, may suffer from performance issues and loss of responsiveness in case of an event flood.

The proposed architecture (see Section 4) introduces several types of FIPA-compliant agents, which implement three levels of sophistication, allowing some events to be processed locally. At the same time, the proposed architecture caters for high availability by utilizing supervision schemes well known in, for instance, telecommunication systems.

To formally model properties of physical models under consideration, a graph-based model (GEM) is proposed (see Section 5). An extension of the model (GEM-CA) is introduced to provide a formal representation including control data.

Further research focuses on control issues within dynamically changing environments. It is driven by recent advances in street lighting technologies including not only multiple power states but also controllable light stream geometries.

References

- [1] Erlang/OTP (Open Telecom Platform), <http://www.erlang.org/>.
- [2] A. Sędziwy and M. Kozieł-Woźniak, Computational support for optimizing street lighting design, *Complex Systems and Dependability* (W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, eds.), vol. 170 of *Advances in Intelligent and Soft Computing*, pp. 241–255, Springer Berlin Heidelberg, 2012.
- [3] A. Mahdavi, Predictive simulation-based lighting and shading systems control in buildings, *Building Simulation*, vol. 1, pp. 25–35, Mar. 2008.
- [4] A. Sędziwy, Effective graph representation for agent-based distributed computing, *Agent and Multi-Agent Systems. Technologies and Applications* (G. Jezic, M. Kusek, N.-T. Nguyen, R. J. Howlett, and L. C. Jain, eds.), vol. 7327 of *Lecture Notes in Computer Science*, pp. 638–647, Springer Berlin Heidelberg, 2012.
- [5] Java Agent Development (JADE), <http://jade.tilab.com>.
- [6] L. Kotulski, On the control complementary graph replication, *Models and methodology of system dependability* (e. a. Jacek Mazurkiewicz, ed.), vol. 1 of *Monographs of System Dependability*, pp. 83–95, Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2010.
- [7] A. E.-D. Mady, M. Boubekeur, G. Provan, C. Ryan, and K. Brown, Intelligent hybrid control model for lighting systems using constraint-based optimisation, *Soft Computing Models in Industrial and Environmental Applications, 5th International Workshop (SOCO 2010)* (E. Corchado, P. Novais, C. Analide, and J. Sedano, eds.), vol. 73 of *Advances in Intelligent and Soft Computing*, pp. 249–259, Springer Berlin Heidelberg, 2010.
- [8] F. Kaku, M. Miki, T. Hiroyasu, M. Yoshimi, S. Tanaka, T. Nishida, N. Kida, M. Akita, J. Tanisawa, and T. Nishimoto, Construction of Intelligent Lighting System Providing Desired Illuminance Distributions in Actual Office Environment, *Artifical Intelligence and Soft Computing* (L. Rutkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, eds.), vol. 6114 of *Lecture Notes in Computer Science*, pp. 451–460, Springer Berlin / Heidelberg, 2010.
- [9] M. Akita, M. Miki, T. Hiroyasu, and M. Yoshimi, Optimization of the Height of Height-Adjustable Luminaire for Intelligent Lighting System, *Artifical Intelligence and Soft Computing* (L. Rutkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, eds.), vol. 6114 of *Lecture Notes in Computer Science*, pp. 355–362, Springer Berlin / Heidelberg, 2010.

- [10] S. Fan, C. Yang, and Z. Wang, Automatic Control System for Highway Tunnel Lighting, *Computer and Computing Technologies in Agriculture IV* (D. Li, Y. Liu, and Y. Chen, eds.), vol. 347 of *IFIP Advances in Information and Communication Technology*, pp. 116–123, Springer Boston, 2011.
- [11] L. Guo, M. Eloholma, and L. Halonen, Intelligent road lighting control systems, tech. rep., Helsinki University of Technology, Department of Electronics, Lighting Unit, 2008.
- [12] I. Wojnicki and L. Kotulski, Controlling complex lighting systems, *Complex Systems and Dependability* (W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, eds.), vol. 170 of *Advances in Intelligent and Soft Computing*, pp. 305–317, Springer Berlin Heidelberg, 2012.
- [13] A. Sędziwy and L. Kotulski, Solving large-scale multipoint lighting design problem using multi-agent environment, *Key Engineering Materials* (D. Su, K. Xue, and S. Zhu, eds.), vol. Advanced design and manufacture IV, pp. 179–182, 2011.
- [14] A. Sędziwy, Representation of objects in agent-based lighting design problem, *Complex Systems and Dependability* (W. Zamojski and et al., eds.), vol. 170 of *Advances in Intelligent and Soft Computing*, pp. 209–223, Springer Berlin Heidelberg, 2012.
- [15] A. Sędziwy, L. Kotulski, and M. Szpyrka, Formal methods supporting agent aided smart lighting design, *Complex Systems and Dependability* (W. Zamojski and et al., eds.), vol. 170 of *Advances in Intelligent and Soft Computing*, pp. 225–239, Springer-Verlag, 2012.
- [16] M. Szpyrka, Design and analysis of rule-based systems with adder designer, *Knowledge-Driven Computing: knowledge engineering and intelligent computations* (C. Cotta, S. Reich, R. Schaefer, and A. Ligeza, eds.), vol. 102 of *Studies in Computational Intelligence*, pp. 255–271, Springer-Verlag, 2008.

Emergence of Collective Escaping Strategies in Caribou Agents

Kun TIAN^a, Ivan TANEV^a, Katsunori SHIMOHARA^{a,1}

^a*Graduate School of Engineering, Doshisha University, Kyoto, Japan*

Abstract. We propose a wolf-caribou predator-prey system to verify our hypothesis that relatively complex collective escaping behavior may emerge from simple, implicit, locally defined, and therefore scalable interactions between the caribou (prey) agents. Proposing two different communication mechanisms – (i) simple, basic mechanism of implicit interaction, and (ii) explicit communications promoting the awareness of the caribou about the identity of the chased one (i.e., empathy), we present a comparative analysis of the implications of these communication mechanisms on the efficiency of evolution of the emerged collective behavior. We used strongly typed genetic programming with exception handling capabilities to evolve the collective behavior of caribou agents. The experimental results suggest that the empathy facilitates the evolution of collective escaping behavior of the team of caribou agents..

Keywords. emergence, multi agent systems, strongly-typed genetic programming.

1. Introduction

Over the past few years, multi-agent systems (MAS) have become more and more important in many aspects of computer science such as distributed artificial intelligence, distributed computing systems, robotics, artificial life, and so forth. MAS introduce the issue of collective intelligence and of the emergence of behavior through interactions between the agents. An agent is a virtual entity that can act, perceive the proximity of its environment and communicate with others; it is autonomous and has abilities to achieve its objectives. MAS contain a world (environment), entities (agents), relations between the entities, a way the world is perceived by the entities, a set of operations that can be performed by the entities and the changes of the world as a result of these actions.

Currently, the main application areas of MAS are problem solving, simulation, collective robotics, software engineering, and construction of synthetic worlds [1]. Considering the latter application area and focusing on the autonomy of agents and the interactions that link them together [2], the following important issues can be raised: What is minimum amount of perception information needed to agents in order to perceive the world? How can agents cooperate? What are the methods, and what are the lower bounds of communications, required for them to coordinate their actions?

¹ Corresponding Author: Katsunori Shimohara, Doshisha University, 1-3 Tatara-Miyakodani, Kyotanabe, Kyoto 610-0321, Japan; E-mail: kshimoha@mail.doshisha.ac.jp

What is the architecture they should feature so that they can achieve their goals? What approaches can be applied to automatically construct the agents' functionality, with the quality of such a design being competitive to the design handcrafted by human?

These issues are of special interest, since the aim is to create MAS which is scalable, robust, flexible, and able to automatically adapt to changes. These features of MAS are believed to be particularly important in real world applications where the approaches to construct synthetic worlds can be viewed as a practical methods, a techniques towards creating complex "situational aware" multi-computer, multi-vehicle, or multi-robot systems based on the concepts of agents, communication, cooperation and coordination of actions [3].

Within considered context, the objective of our research is an automatic design of autonomous agents which situated in inherently cooperative environment are capable of accomplishing complex tasks through interaction. The social behavior, needed to accomplish the complex task, might emerge in MAS from relatively simply defined interactions between the agents.

The predator-prey pursuit problem (PPPP) was introduced by Benda et al [4] and comprised four predator agents whose goal is to capture a prey agent by surrounding it on four sides in a grid-world. This problem has been used to study phenomena such as competitive co-evolution [5][6][7], multi-agent strategies, and multi-agent communication.

In this work we propose a wolf-caribou system as a reverse PPPP system, which emphasis on the importance of the collective strategy of inferior (i.e., slower) prey agents (caribou) for successful escape from a superior (faster) predator agent (wolf).

2. Proposed Approach

The wolf-caribou predator prey-problem (WCPPP) consists of two subsystems: the (i) Genetic Programming (GP) manager and the (ii) other simulated world, respectively. The former keeps track of the evolved IF-THEN behavioral rules of caribou agents (represented as "genetic programs" in GP), and performs the main genetic operations – selection, crossover, and mutation. The fitness evaluation phase of GP, on the other hand, is implemented by the latter subsystem – the simulated world. It models the behavior of the entities (wolf and caribous) in WCPPP as a multi-agent system according to the currently evaluated IF-THEN behavior rules, and assigns the fitness values to the team of caribou agents as the time needed for the wolf to capture a caribou. The result of the simulation of the world – the fitness value – is returned to the GP manager subsystem at the end of evaluation phase.

GP, employed in our work to evolve the collective behavioral rules of escaping caribou agents, is a domain-independent problem solving approach in which a population of computer programs (individuals) is evolved to solve problems [8]. The simulated evolution in GP is based on the Darwinian principle of reproduction and survival of the fittest. In GP genetic programs (individuals) can be represented as parsing trees whose nodes are functions, variables or constants. The nodes that have sub-trees are non-terminals they represent functions where the sub-trees represent the arguments to function of that node. Variables and constants are terminals they take no arguments and they always are leaves in the parsing tree. The set of terminals for evolving agent's behavior includes the perceptions (stimuli), and the actions (response) which the agent is able to perform. The set of functions comprises the arithmetical and

logical operators, and the IF-THEN function, establishing the relationship between certain stimulus and corresponding response(s) [9]. The main parameters of GP are shown in Table 1.

3. Modeled MAS for WCPPP

We considered a configuration with one superior (i.e., faster) wolf (predator) agent and eight inferior (slower) caribou (prey) agents. The successful escape by the caribous corresponds to the situation when the wolf is unable to capture any caribou during the limited time of the trial. In order to obtain more general collective behavior of the caribou, the trial consists of ten (zero to nine) different initial situations, with different initial positions and initial orientations of the caribou agents. In all ten considered situations, the wolf is positioned in the center of the world.

Table 1. Main parameters of genetic programming

Parameter	Value
Population size	400
Selection rate	10%
Mutation rate	1%~80%
Elitism	4%
Termination criteria	((Fitness>600) AND (Successful situations=9)) OR (No fitness improvements for 60 generations)

Fitness value is defined as the time needed for the wolf to capture a caribou. The maximum (i.e., the best) fitness value would correspond to the number, equal to the total number of the time steps of the trial (in the considered case – 600).

We adopted the subsumption architecture for the caribou agents, consisting of the three simple modules, responsible for three basic behaviors, as illustrated in Figure 1. The caribou implements an *escaping behavior* if the wolf is within its range of visibility and no other caribou is seen. The escaping speed corresponds to the maximum speed of caribou, and the escaping direction is straight away from the wolf. We should note that, due to the inferiority of the caribou agents, inherently, this behavior could not guarantee a successful escape from the wolf. Caribou agents exhibit an evolved *collective escaping behavior* (shown in Figure 1 as “distracting”) in order to coordinate their movement in a way that could mislead, or deceive the wolf for a time, long enough to prevent the latter from capturing any caribou during the trial. Finally, according to the proposed architecture of the caribou agents, caribou agents could exhibit a *wandering behavior* if no other entity (wolf nor other caribou) is within its visible range. Both the caribou and wolf agents feature a *limited amount of stamina*, in that their respective maximum speeds decrease with the increase of the distance they have traveled during the trial.

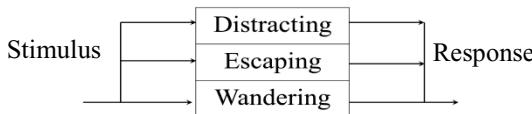


Figure 1. Subsumption architecture of caribou agents, consisting of three simple behaviors: *wandering* (lowest priority), *escaping*, and evolved (as IF-THEN behavioral rules) *distracting* – a collaborative escaping behavior (highest priority).

The caribou agents are inferior in that (i) they feature shorter ranges of visibility and (ii) lower maximum speed than those of the wolf. These conditions render the task of escaping from wolf impossible unless the caribou agents employ a collective escaping strategy. The main parameters of the simulated entities in WCPPP are shown in Table 2. The perceptions of caribou agents are listed in Table 3. As shown in the table, in order to verify the importance of the awareness of the identity of chased caribou (i.e., empathy) we introduced the following four perceptions: Chased_Peer_d, Chased_Peer_a, Speed_Chased, and Faster_than_Chased. A sample evolved IF-THEN rule governing the behavior of caribou agents is illustrated in Figure 2.

Table 2. Main parameters of simulated entities

Parameter	Wolf	Caribou
Number	1	8
Diameter	68mm	78mm
Range	900mm	660mm
Max speed	19mm/s	17mm/s

Table 3. Perception parameters of caribou agents

Parameter	Type	Interpretation
Wolf_d	Integer	Distance to the wolf
Wolf_a	Integer	Bearing (angle in the visual field) of the wolf
Speed_Wolf	Integer	Speed of wolf
Peer_d	Integer	Distance to the closest caribou
Peer_a	Integer	Bearing (angle in the visual field) of the closest caribou
Speed	Integer	Own speed
Speed_Peer	Integer	Speed of the closest caribou
Chased_Peer_d	Integer	Distance to the chased caribou
Chased_Peer_a	Integer	Bearing (angle in the visual field) of the chased caribou
Speed_Chased	Integer	Speed of the chased caribou
Chased	Boolean	<i>True</i> if caribou is the chased one, <i>False</i> otherwise
Faster_than_Chased	Boolean	<i>True</i> if own speed is higher than the chased caribou; <i>False</i> otherwise

```

if (Speed >= Speed_Wolf)
then
begin
    Turn(Chased_Peer_a);
    Go_1.0;
end;

```

Figure 2. Sample evolved IF-THEN rule governing the behavior of caribou (prey) agents

4. Experimental Results

4.1. Effect of Empathy on the Efficiency of Evolution

In order to identify the importance of empathy of caribou agents on both the efficiency of evolution of their collective behavior, and on the efficiency of such a behavior, we assumed two experimental conditions:

- The caribou agents *do not care* about the identity of the chased caribou (i.e., no empathy), and
- The caribou agents *are aware* of the identity of the chased caribou (i.e., with empathy).

For both cases, we implemented 20 evolutionary runs of genetic programming. The dynamics of the number of successful initial situations (i.e., situations where caribou agents escape successfully from the wolf) for both cases are shown in Figure 3. As Figure 3 illustrates, modeling the empathy facilitates the evolution of escaping behavior of caribou agents. Indeed, the empathy is associated with higher and faster achieved values of the number of successful situations per generation.

Figure 4 illustrates the number of successful by the end of 20 independent runs *with-* (grey bars in Figure 4) and *without* (white bars in Figure 4) the modeling of the empathy. As shown in the figure, the empathy contributes to the achievement of all ten successful situations in 50% of the evolutionary runs, while, without empathy, only 30% of evolutionary runs are successful.

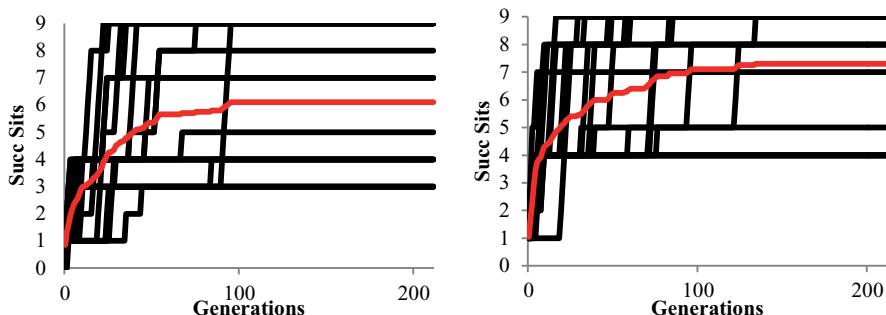


Figure 3. Number of situations (per generation) successfully cleared by the team of caribou agents *without* (left) and *with* (right) empathy, respectively. Results are obtained from 20 evolutionary runs of genetic programming

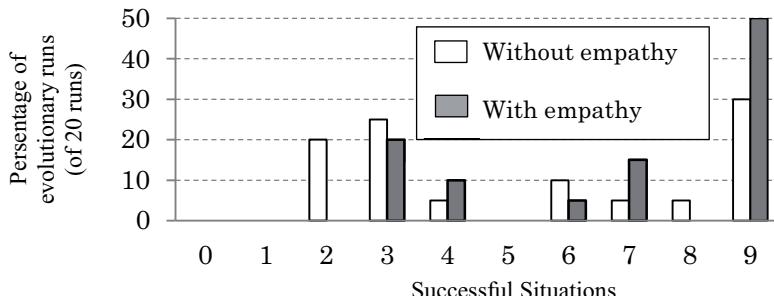


Figure 4. Percentage of evolutionary runs (of 20 runs) vs. number of successful situations for the cases of caribou agents *without-* and *with* empathy, respectively.

4.2. Emergent Escaping Behaviors

For the experimental conditions with the modeling of empathy, we observed an emergence of two collective, escaping behaviors. These behaviors are elaborated below. Considering these emergent collective escaping behavior of caribou agents, we could conclude that the empathy is important not only for the efficiency of the evolution of the escaping strategies (as shown above), but also for the efficiency of the successfully evolved escaping strategies. Awareness about the identity of the currently chased caribou is very important for the implementation of the emergent behaviors, mentioned below. Moreover, the empathy, or, more precisely, the lack of it, also suggest the caribou agents that they can *rest and save the precious stamina* when no chased caribou is seen.

4.2.1 Awareness Behavior

Awareness behavior as shown in Figure 10 indicates that if distant caribou agents sense the wolf, they will escape from it well before they immediately sense the latter. This kind of awareness allows the caribou to actually extend the range of their sensors by means of collaboration. This, in turn, gives more time to the caribou agents to respond to the presence of the wolf well before actually being able to sense the presence of the latter. Sometimes, this kind of behavior can be seen in animal world when one member of a group warns the group about an imminent danger and then sacrifices itself for the survival of the group as a whole.

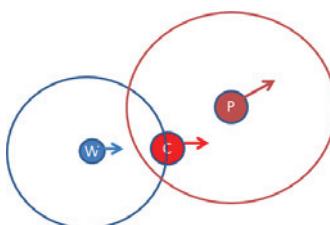


Figure 5. Awareness behavior: Despite that wolf (W) is not in visible for the peer caribou agent (P), the latter can still sense *indirectly* the presence of the wolf by receiving information from the chased caribou (C). The caribou (P) can respond to the presence of wolf well in advance. Co-worker Behavior

4.2.2 Deceptive Collaborative Behavior

Similarly to the behaviors observed in animal world, deceptive action emerged in the proposed WCPPP when caribou agents collaborate in order to disturb (or, in other words, deceive) the wolf by constantly changing their roles. Because wolf always chases the closest target, peers can change their locations in order to prompt the wolf to switch its attention to the different (closest) caribou. The resulting zigzag movement features lower overall speed despite the speed superiority of the wolf. This behavior is illustrated in Figure 6.

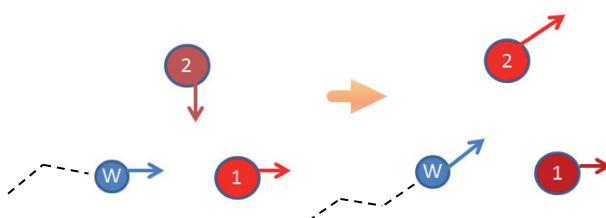


Figure 6. Deceptive collaborative behavior: caribou #2 senses that caribou #1 is chased by wolf (W), and tries to “attract” the wolf’s attention by coming closer to it (left). As a result, wolf turns to chasing caribou #2 (right). Caribou #1 and #2 then alternate their roles until the wolf runs out of stamina and out time of trial without capturing either of them.

5. Conclusion

We presented the result of our work on use of genetic programming for evolving collective escaping behavior of caribou agents in wolf-caribou problem. We proposed the wolf-caribou problem as a reversed case of predators-prey pursuit problem, to verify our hypothesis that relatively complex collective escaping behavior of inferior caribou agents may emerge from simple, implicit, locally defined, and therefore effective and highly-scalable interactions between the caribou agents.

We used genetic programming in order to evolve the collective escaping behavior of caribou agents. The behavior is encoded as evolvable IF-THEN rules. The results of evolution of collective escaping behavior of caribou demonstrate the emergence of two interesting behaviors: *awareness* and *deception*, respectively. The first one allows the caribou to actually extend their perceptual range by warning about the presence of wolf well before they sense it by their own. The second behavior allows the caribou to alternate their relative position to the wolf, thus prompting it to switch its chasing behavior to different targets. With empathy modeled as awareness of the identity of the chased caribou, the team of inferior caribou agents can effectively escape from the chase of the superior wolf.

In the future, we are planning to investigate the influence of the various number of agents, differences between the speeds of the wolf and caribous, and various capabilities or relationships introduced in caribou agents, on emergence of escaping strategies. We consider the defensive military and crime prevention as possible application areas of the proposed research.

References

- [1] Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Harlow: Addison Wesley Longman, 1999.
- [2] Parunak, H. Van D., Brueckner, S., Fleischer, M., Odell, J.: Co-X: Defining what Agents Do Together, Proc. of the AAMAS 2002 Workshop on Teamwork and Coalition Formation, Onn Shehory, Thomas R. Ioerger, Julita Vassileva, John Yen, eds., Bologna, Italy, 2002.
- [3] Tanev, I and Shimohara, K.: On Role of Implicit Interaction and Explicit Communications in Emergence of Social Behavior in Continuous Predators-prey Pursuit Problem, Proc. Of Genetic and Evolutionary Computation, pp.74-85, 2003.
- [4] Tanev, I and Shimohara, K. : XGP : XML based Genetic Programming Framework, Proc. of the 34th Symposium of the SICE on Intelligent System, pp.183-188, 2007.
- [5] Nolfi S. and Floreano D.: Coevolving predator and prey robots: Do "arms races" arise in artificial evolution? Artificial Life, 4(4), pp.337–357, 1998.
- [6] Savill N.J. and Hogeweg P.: Evolutionary stagnation due to pattern-pattern interactions in a co-evolutionary predator-prey model, Artificial Life, 3(2), pp.81–100, 1997.
- [7] Haynes T. and Sen S.: Evolving behavioral strategies in predator and prey. IJCAI-95 Workshop on Adaptation and Learning in Multi-agent Systems, 1995.
- [8] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA, MIT Press , 1992.
- [9] Tian, K., Tanev, I and Shimohara, K. : Emergence of Collective Escaping Strategies of Caribou Agents in Wolf-caribou Predator-prey Problem, the 8th Int. Conf. on Humanized Systems, pp.220-223, 2012.

On Acceleration of Multi-Agent System Performance in Large Scale Photometric Computations

Adam SĘDZIWy^{a,1}

^a AGH University of Science and Technology, Kraków, Poland

Abstract. Multi-agent computations are a useful computing paradigm applied in various areas, such as smart grids or distributed information processing. Another field of their application is outdoor lighting design, which is characterized by highly time-consuming optimization tasks. In this paper we focus on the latter case, aiming at scalability and reducing the computing time, which are crucial for solving the lighting design problem. To improve the efficiency of parallel, agent-based computations the results, a reuse approach is introduced. A quantitative comparison with other approaches is also included.

Keywords. graph, slashed graphs, distributed computing, multi-agent system, lighting control

1. Introduction

Novel technologies related to large-scale distributed computations emerge together with the growing size of distributed data such as Internet resources and the corresponding demand for their processing. Another example of such demanding applications are smart grids, which are compounds of numerous, distributed infrastructure components (e.g., smart meters, sensors and so on) and data warehouses holding the information (also historical) about the state and activity of the entire system. In the second case, one may deal with an additional demand, which is on-line operational responsiveness.

In this article, we consider the problem of outdoor lighting design [1]. The key issue of the street lighting design process is setting the distribution and configuration of fixtures in a way which guarantees minimal power consumption and satisfies compulsory lighting standards. It should be stressed that those objectives are also crucial in a fixture replacement task. One example is migration from a high-pressure discharge lamp-based illumination to a LED-based one, which may be carried out in an entire city area. This problem is a variant of non-linear multidimensional and multiobjective optimization. In this paper, we focus on a method of reducing the time required to complete photometric computations, which include multidimensional and multiobjective optimization, performed in the scale of a city (thousands of streets).

¹Corresponding Author: Adam Sędziwy, AGH University of Science and Technology, Department of Applied Computer Science, al. Mickiewicza 30, 30-059 Kraków, Poland; E-mail: sedziwy@agh.edu.pl.

The first step towards a scalable computing method is finding a proper formal model describing the environment being considered. Such a formal model has to support *simple* fragmentation (i.e., one which doesn't cause a system overload), which is prior to processing parallelization. Another important property of a model, which may be not obvious at a high level of abstraction of its specification is the complexity of coordination in a distributed system (a multi-agent system in the considered case), implied by a fragmentation method. Such complexity may generate a significant overhead in communications or cause long response times. Two formalizations which meet those requirements are graph-based models using fragmentation approaches referred to as *replicated complementary graphs* (RCG) (see [2,3]) and *slashed representation* [4]. Due to the better performance related to the coordination, the second approach was selected.

The next step made after selecting a suitable formal model is its decomposition and deployment of a multi-agent system in order to achieve a final goal, i.e. optimization of a lighting system. Processing parallelization reduces the computing time *per se*. The additional time reduction may be achieved by means of the *result reuse method*, which is introduced in this paper (Section 4.2). It should be emphasized that the proposed approach may be applied successfully to other high complexity problems, in particular those which are not represented by graphs.

The paper is organized as follows. In Section 2 we present works concerning the application of the result reuse method in various types of problems. The concept of *slashed graphs*, which are used as a modeling framework for the presented test case, is sketched briefly in Section 3. Properties of a considered multi-agent system are presented in Section 4. Section 5 contains the description of performance tests. Final conclusions are included in Section 6.

2. Related Works

To improve the efficiency of large-scale computations, new enhancements are introduced. One example is parallel data retrieval algorithm [5]. However, this concept aims at improving communications performance rather than at boosting computation speed. Another concept, focused just on improving computations performance, is implemented by existing solutions like Google's MapReduce [6] programming model and related tools, e.g., the Hadoop framework [7]. It relies on reusing results obtained in the computation life cycle. This approach is used in a range of applications.

Result reuse underlies ReStore, the system which manages the storage and reuse of results in distributed large data processing [8]. In this approach, intermediate results of processed queries are stored and reused in subsequent ones. Another case of result reuse is implementation of field-programmable gate-arrays (FPGAs), used in signal recognition systems, presented in [9]. The concept of result reuse was also introduced for morphological operations, in the technique of the partial-result-reuse architecture [10]. In all those cases either resource demand or time complexity reductions have been achieved.

The method introduced in this paper exploits the result reuse concept in the context of multi-agent-based large-scale computations.

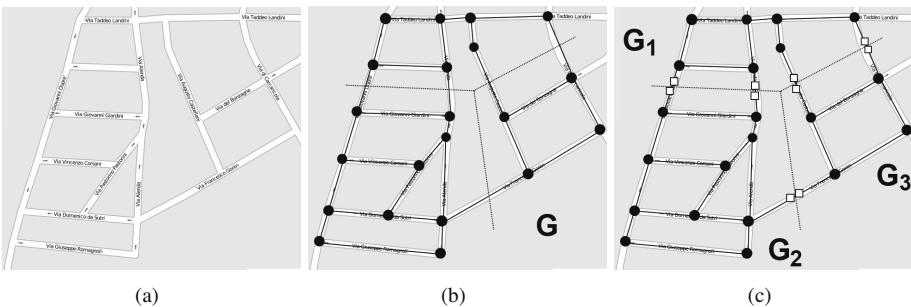


Figure 1. (a) City map (b) Graph representation – G (c) Slashed representation – G

3. Slashed Representation of a Graph

The slashed form of a graph was introduced in [4] to reduce coupling among subgraphs obtained through decomposition of a centralized graph, and thereby to simplify operations performed by a multi-agent system (MAS).

As shown in [11], graphs are a suitable representation for urban spaces, which enables scalable, agent-based computations. Since a formal model ensuring effective distributed computations is required, the slashed graph representation, discussed in detail in [4], is the most convenient. In this model, a graph edge represents a street and edge attributes hold its physical properties like length or width. Graph slashing may be accomplished by logical decomposition of an underlying map in points determined by structural properties (e.g., intersections or points where a single carriageway turns into a dual one).

Decomposition of a centralized graph into its slashed form prepares an environment for deployment of a multi-agent system.

Figure 1(a) shows an exemplary scene – an area of a city. The graph representation G (centralized) of the scene is shown in Figure 1(b): the dotted lines mark boundaries of the further decomposition. Figure 1(c) contains the **slashed** (i.e., decomposed) **form** of graph G , denoted as G . Subgraphs of G (in the considered case G_1, G_2, G_3) are referred to as **slashed components** of G . Newly appeared vertices in G , marked as empty squares in Figure 1(c), obtained as a result of edge slashing, are referred to as **dummy nodes**. All vertices marked as filled circles in Figure 1(c), existing previously in the centralized graph G , are called **core nodes**.

For the slashed graph-based approach, one agent is ascribed to exactly one slashed component. This approach enables achieving low complexity of coordination in operations performed by agents, e.g., when shifting a border (dotted line in Figure 1(c)) among slashed components.

4. Multi-Agent System Properties

Using a multi-agent system for photometric communications considered here is necessary due to the decomposition problem: setting up a load-balanced decomposition of a main computational task, with a minimal number of interfaces among sub-tasks, is complex enough to significantly reduce the efficiency of conventional parallel (non-agent)

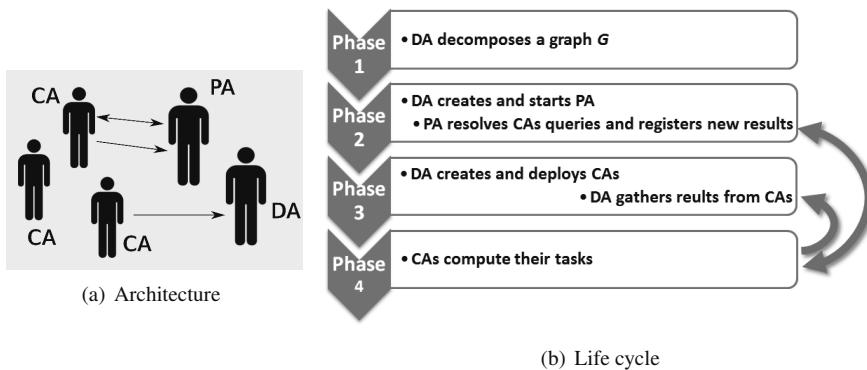


Figure 2. A multi-agent system performing computations

computations. On the contrary, agent-based methods of graph decomposition have been developed for several years (e.g., see [12])

In this section, we present key aspects determining properties of the considered multi-agent system: architecture, logic of the reuse-supported computing and the performance issues.

4.1. MAS Architecture

For the considered computations, we introduce the MAS architecture (Figure 2 (a)) supported on three types of agents: *dispatcher agent* (DA), *pattern agent* (PA) and *computing agent* (CA).

- The *dispatcher agent* is responsible for creating all remaining agents and for gathering the results of computations. Only one DA is present in a multi-agent system.
- The *pattern agent* holds a registry containing all known results and their corresponding patterns. Its role is resolving CAs' queries for required results and registering new ones. Only one PA is present in a multi-agent system.
- The *computing agent* is responsible for making photometric computations and registering results in a registry held by a PA. In a general case, multiple CAs are present in a multi-agent system.

In the initial phase of a multi-agent system life cycle (see Figure 2 (b)), only a DA is present. It decomposes a centralized graph into slashed components which are to be maintained by CAs. In the next phase, a DA creates a pattern agent which manages the patterns/results registry. In the third phase, since a PA is operational, a DA may create and deploy computing agents on particular slashed components. Thus, one CA has a single slashed component assigned. In the fourth phase CAs perform their computations. When they are completed, a CA sends results to a DA and terminates. The detailed description of a computing agent activity is presented in Subsection 4.2. When all streets are processed, a PA terminates.

4.2. The Computation Reuse Method. Computing an Agent's Life Cycle

The result reuse method is based on the observation that for large urban spaces, some computation areas, referred to as *patterns*, repeat. In this context, the *pattern* is under-

stood as the minimal information about the considered road required to perform photometric computations. In a trivial case, the pattern may be a pair of the form $p = (l, w)$ containing the length and width of a given straight section of a road. Another example is the pattern $p' = (l, w, \Delta)$, where Δ denotes street lamp spacing.

It is assumed that all patterns and corresponding computation results are stored in the registry M . Initially, M is empty; once a multi-agent system starts, M is successively filled with results provided by CAs (see Algorithm 1, line 11).

If a pattern p repeats in any slashed component and was previously processed by another CA, then a relevant result may be obtained by performing registry lookup only (see line 8 of Algorithm 1). Otherwise, a CA performs the computations for p . After calculating a result for a given pattern, a computing agent sends it to a PA which maintains the registry M .

A generic schema of CA's ComputeBehaviour is presented in the form of a pseudocode in Algorithm 1

Algorithm 1: ComputeBehaviour(G)

```

input :  $G$  – slashed component to be processed
output:  $T[G]$  – table of results for all streets represented in  $G$ 
1 begin
2   const QuerySize;
3   Empty the message queue;
4    $S \leftarrow$  list of streets represented in  $G$ ;
5    $\Pi \leftarrow$  patterns of streets in  $S$ ;
6    $P \leftarrow$  set of at last QuerySize-element lists of patterns from  $\Pi$ ;
7   foreach List  $q \in \Pi$  do
8     Send QUERY( $q$ ) to PA for results for particular patterns;
9     if response timeout occurred or some results are missed then
10       Compute results  $r$  for non-matched patterns  $p$ ;
11       Send REGISTER([ $p, r$ ]) to PA;
12   Associate result with each street in  $S$ ;
13   Send REGISTER( $T[G]$ ) to DA;

```

Initially, a computing agent determines all patterns present in its slashed component. It is obvious that the number of patterns is not greater than the number of streets. Next, the CA enters the cyclic phase of the alternating steps:

1. CA removes outdated responses from the message queue (Algorithm 1, line 3).
2. CA sends a query to a pattern agent for results matched to particular patterns.
3. CA receives a response (in blocking receive with timeout – Algorithm 1, line 9) from the PA. If no message is received (due to the timeout) or some patterns could not be matched with the results, a CA performs computations for those non-matched patterns.
4. CA sends a message to a PA to register all results obtained for non-matched patterns in M .

The number of passes through these steps (i.e. a number of queries sent to a PA) depends on the size of the slashed component, but also on the `QuerySize` constant which is equal to the number of patterns sent in a single query.

Algorithm 2: ServiceQueryBehaviour

```

1 begin
2    $PV \leftarrow$  pattern-value map;
3   while Not terminated do
4      $m \leftarrow$  receiveMessage();
5     if  $m$  is REGISTER message then
6        $[(p, v)] \leftarrow$  list of pattern-value pairs fetched from  $m$ ;
7        $PV \leftarrow PV \cup [(p, v)]$ ;
8     if  $m$  is QUERY message then
9        $[p] \leftarrow$  list of patterns fetched from  $m$ ;
10       $[r] \leftarrow$  results of  $PV$  lookup made for  $[p]$ ;
11      Send RESPONSE( $[r]$ );
  
```

The algorithm 2 contains pseudocode representing the ServiceQueryBehaviour scheme of a pattern agent. It consists of a loop containing the operation of blocking receive (see line 4) and two actions handling particular types of requests. The loop breaks when a PA is terminated by a dispatcher agent.

4.3. Efficiency Issues

Among factors impacting the efficiency of a multi-agent system a, the value of the `QuerySize` constant is present. Let us consider two extremes.

For `QuerySize` equal to \max where

$$\max = \max_{H \in \mathcal{G}} \{\text{Number of patterns in } H\},$$

a CA sends only one query to the pattern agent. Assuming that all computing agents keep equally-sized slashed components and that they are initiated by a DA in the same moment, the following scenario will occur: a PA receives multiple queries but since no agent has registered any result yet, none of the queries may be resolved. In such a case, agents cannot reuse results. The only benefit for the performance of the system is related to parallel computations.

The second extreme is for `QuerySize` equal to 1. For this case, steps 2-4 described in Subsection 4.2 (see Algorithm 1, line 7) will be repeated N_p times, where N_p is a number of patterns for a given slashed component. Although the pattern/value registry filling speed will be maximal for this case, it generates a communication overhead which may affect the overall performance of a MAS.

To establish the optimal value for `QuerySize`, additional parameter tests have to be made. In Section 5.1, computing times for various `QuerySize` values will be investigated.

The second factor influencing the efficiency of the reuse method is the complexity of an *atomic computation* made by a CA. In this context, the *atomic computation* term refers to the set of operations made on the smallest input data subset (i.e., a single street, in our case), producing a result which is the subject of computations (e.g., the average illuminance). If the duration of an *atomic computation* is low, then the effect related to the reuse is neglectable, or precisely, the communication overhead dominates the achieved time gain.

5. Tests

The object of the tests was a graph $G = (V, E)$ representing a part of the city of Rome, generated from the OpenStreetMap source; $|V| = 1500$, $|E| = 2886$. The graph G was decomposed by a DA into subgraphs $G_i = (V_i, E_i)$ such that $|E_i| \leq E_{max} = 30$. A separate CA was assigned to each of these subgraphs.

The task of each agent was to find a configuration of the lighting infrastructure parameters (e.g. pole height, lamp spacing, dimming and so on) which ensures minimal power consumption and, on the other side, compliance with given road lighting standards. Since the used optimization method was the *brute force* approach, such a task may be easily presented as a sequence of nested loops (see Algorithm 3).

Algorithm 3: Optimize

```

1 begin
2   for  $p_1 \in P_1$  do
3     for  $p_2 \in P_2$  do
4       ...
5       for  $p_m \in P_m$  do
6         ComputeLighting( $p_1, p_2, \dots, p_m$ );
7       ...

```

A value of m is the number of parameters taken into account during optimization. It is assumed that the ComputeLighting routine (see Algorithm 3, line 6) is executed in a constant time (i.e., is $\mathcal{O}(1)$), thus obviously, the complexity of Algorithm 3 is $\mathcal{O}(k^m)$, where $k = \max_{i=1,\dots,m} |P_i| = const$. In tests, a depth of $m = 4$ was assumed.

To investigate the properties and gains provided by the reuse method, three tests were conducted. Their goals were:

1. determining the optimal value of the `QuerySize` parameter (with respect to the computing time),
2. assessing the computing time reduction achieved for particular computational approaches,
3. tracking the utilization of results stored in the registry M .

The tested multi-agent system was implemented using the JADE framework, which is a FIPA-compliant agent environment [13,14].

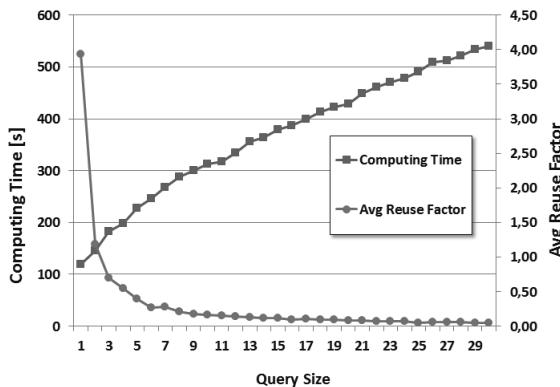


Figure 3. The computing time as a function of *QuerySize*

5.1. Impact of *QuerySize* Parameter on Computing Time

To establish the optimal value of

QuerySize (i.e., one which assures the best MAS performance), we tested computing times for $1 \leq \text{QuerySize} \leq E_{\max} = 30$. The test was repeated $N=10$ times for each *QuerySize* value and the results (computing time, reuse factor) were averaged over those N passes. Figure 3 contains the diagram of obtained results. Besides the computing time, we also investigated the average *reuse factor* (RF), which is defined as the average number of successfully responded queries per pattern, where the average is calculated over all patterns stored in the registry M .

The results show that the minimum computing time is obtained for the single pattern-query policy: *QuerySize* = 1 (see Figure 3). It follows from the fact that the average RF reaches its maximum (3.94) for this *QuerySize* value. The maximum RF is then 83.7, which means that the result for some pattern was successfully requested over 83 times. This causes a reduction of the computing time.

The conclusion of this test is that the time overhead related to inter-agent communication doesn't impact the system performance for *QuerySize*=1. Hence, we select it for further considerations.

5.2. Comparison of Various Computational Approaches

In this test, conducted on the 4 core CPU (Intel Core i7), we compare various computational method to reveal the impact of using the multi-agent approach and, on the other side, the effects of the reuse method. In the test, the following four approaches were applied: centralized computations, agent based computations, centralized computations supported by the reuse method and agent-based computations supported by the reuse method. Figure 4 shows the obtained computing times. It can be seen that the reuse approach significantly reduces the computing times for both the centralized (13.3 times) and the agent-based (5.3 times) approach. The highest acceleration (26.5 times) is achieved when transiting from the simple, centralized approach to the agent-based one with result reuse.

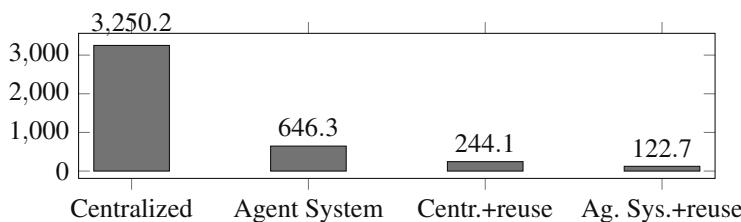


Figure 4. Execution times [sec] for various computational approaches

6. Conclusions

In this paper, we presented a method for reducing the computing time of large-scale photometric computations, which may be regarded as multi-dimensional non-linear optimization. In the introduced approach, we benefit from the synergy of parallel, multi-agent computations performed in an efficient, slashed-graph environment and the result reuse method, which is well applicable to the considered problem.

The conducted tests show that combining both concepts leads to significant time reduction in comparison to sequential processing. The result reuse method reduces the computing time over 13 times for the centralized approach. For agent-based computations, it gives a reduction factor of 5.3. Taking into account the 5.03x acceleration achieved thanks to the parallel processing, we ultimately reduce the computing time for the agent-based, results reuse-supported approach by a factor of 26, when compared to the centralized one.

Although the discussed case and performed tests concerned photometric computations, the proposed methodology may be also applied to other problems where the time overhead is implied, e.g., by queries sent to a data warehouse.

Acknowledgments

The work is co-financed by the European Union, Human Capital Operational Programme, SPIN project no. 502.120.2066/C96, the Research Fund no. 11.11 120.859 and from the resources of Alive & KIC-ing project.

References

- [1] A. Sędziwy and M. Kozień-Woźniak, Computational support for optimizing street lighting design, *Advances in Intelligent and Soft Computing*, vol. 170, pp. 241–255, 2012.
- [2] L. Kotulski, Distributed graphs transformed by multiagent system, *Lecture Notes in Computer Science*, vol. 5097, pp. 1234–1242, 2008.
- [3] L. Kotulski, GRADIS - Multiagent Environment Supporting Distributed Graph Transformations, *Lecture Notes in Computer Science*, vol. 5103, pp. 644–653, 2008.
- [4] A. Sędziwy, Effective graph representation for agent-based distributed computing, *Lecture Notes in Computer Science*, vol. 7327, pp. 638–647, 2012.
- [5] K. Jinoh, Data parallelism for large-scale distributed computing, *Int'l Journal on Internet and Distributed Computing Systems*, vol. 1, 2011.
- [6] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, *Commun. ACM*, vol. 51, pp. 107–113, Jan. 2008.

- [7] M. Farhan Husain, P. Doshi, L. Khan, and B. Thuraisingham, Storage and retrieval of large RDF graph using hadoop and mapreduce, *Proceedings of the 1st International Conference on Cloud Computing*, vol. 5931, pp. 680–686, 2009.
- [8] I. Elghandour and A. Aboulnaga, ReStore: reusing results of MapReduce jobs, *Proc. VLDB Endow.*, vol. 5, pp. 586–597, Feb. 2012.
- [9] M. Demertzi, P. Diniz, M. Hall, A. Gilbert, and Y. Wang, The potential of computation reuse in high-level optimization of a signal recognition system, *Parallel and Distributed Processing, 2008. IPDPS 2008.*, pp. 1–5, 2008.
- [10] S.-Y. Ch., S.-Y. M., and L.-G. Ch., Partial-result-reuse architecture and its design technique for morphological operations, *Acoustics, Speech, and Signal Processing, 2001. Proceedings.*, vol. 2, pp. 1185–1188, 2001.
- [11] A. Sędziwy, Representation of objects in agent-based lighting design problem, *Complex Systems and Dependability* (W. Zamojski, ed.), vol. 170 of *AISC*, pp. 209–223, Springer Berlin Heidelberg, 2012.
- [12] L. Kotulski and A. Sędziwy, Agent framework for decomposing a graph into the equally sized subgraphs, *Proceedings of the 2008 International Conference on Foundations of Computer Science, FCS 2008*, pp. 245–250, 2008.
- [13] Foundation for Intelligent Physical Agents (FIPA), <http://www.fipa.org>.
- [14] Java Agent Development (JADE), <http://jade.tilab.com>.

Agent-Based Modeling and Simulation

Modeling Value Co-creation Process in Complex Service Systems Using Kauffman's NKCS Architecture

Chathura RAJAPAKSE ^{a,1} and Takao TERANO ^b

^{a,b} Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Japan

Abstract. Despite its extensive use in emerging Service Science research, value co-creation in complex service systems still remains highly conceptual and ambiguous. Therefore, there exists a need of tools and techniques to model complex service systems making the process of value co-creation operational. Agent-Based Modeling and Simulation has been recognized as a powerful technique in studying complex adaptive systems and hence, becomes a potential approach in studying complex value co-creation interactions and emerging properties of service systems. However, the computational representation of the value co-creation process among agents of a complex service system still remains challenging. This paper proposes a candidate method to address this issue based on Kauffman's NKCS theoretical framework that mimics the co-evolution of multiple species in a biological ecosystem. The key aspects of value co-creation have been discussed using a toy example of two agents. Furthermore the extendibility of the method to model complex service systems has also been discussed.

Keywords. Agent-Based Modeling, Value Co-creation, Complex Service Systems, NKCS Model, Service Science, Management and Engineering (SSME), Service-Dominant Logic

1. Introduction

Recently, many researchers have seen understanding complex service systems as an urgent need in order to address bigger issues in societies built on such systems [1], [2], [3]. In complex service systems, each interconnected individual or an institution render their competency, i.e. service, within the system to co-create value and achieve individual and collaborative systemic outcomes, creating a whole that is bigger than the sum of its parts [1]. Having its roots in Service-Dominant logic [4], the notion of value co-creation insists that value cannot be added beforehand and exchanged for something (usually money) but rather has to be co-created at the time of use by the customer. However, the term value co-creation is highly conceptual and ambiguous, despite it is being used extensively in the literature without criticizing or even without questioning [5]. According to [6], the notion of co-creation appears to be tricky and difficult to

¹ Corresponding Author: Chathura Rajapakse, Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, 4259, Nagatsuta-cho, Midori-ku, Yokohama, Japan; E-mail: chathura@trn.dis.titech.ac.jp.

define, and is often analyzed neither recognizing its systemic nature nor by adopting a systems thinking approach. Thus, despite being recommended as early as in 2008 by [2], we recognize a research gap in developing tools and techniques to operationalize the value co-creation phenomenon in complex service systems.

Agent-based modeling and simulation [7] has been acknowledged for its potential as a technique for studying complex adaptive systems [8], [9]. Usually, complex adaptive systems resist analysis by decomposition and the complexity imposes limitations on the ability of human beings to understand such systems without the assistance of computer simulations. Complex service systems, for example markets, exhibit features of complex adaptive systems making Agent-based modeling a candidate approach in studying them [10]. However, the process of value co-creation is central to the study of complex service systems and hence, in our opinion, the computational representation of the value co-creation process among agents of a complex service system model is a challenging task.

In this paper, we introduce the NKCS theoretical framework of Kauffman [11] that mimics the co-evolution of multiple species in a biological ecosystem, as a candidate method to represent the value co-creation process in complex service systems. We use a toy example of two interacting agents, a service provider and a customer, to formalize the key concepts and show how their satisfaction (value in use) co-evolves. We further discuss how this can be extended to a context of a complex service system that involves a value constellation [12]. The rest of the paper is organized as below. Section 2 revisits the literature on complex service systems, value co-creation and agent-based modeling whereas Section 3 introduces Kauffman's NKCS model. Section 4 presents the details and the outcome of our agent-based model with two agents while Section 5 presents a discussion on the model and how it can be extended to a complex service system context. Section 6 provides the concluding remarks.

2. Literature Review

2.1. Complex Service Systems

According to [13], service science aims to explain and improve interactions, in which multiple entities work together to achieve win-win outcomes or mutual benefits. This leads to the notion of service systems, in which multiple entities work as a system, offering services to each other through dynamic interactions. More precisely, a service system is defined as value co-creation configurations of people, technology, value propositions connecting internal and external service systems and shared information such as languages, laws, measures and methods [3]. The patterns of interactions among multiple entities in a system such as providers, customers, competitors and authorities as well as all the value propositions that connect these entities determines the complexity of the service system [13]. The extant research views complexity in two ways [6]. First, complexity is essentially associated with increasing interconnections and interdependence among people, businesses and nations in the globalized world. Second, when the focus shifts to service interactions to look for (and understand) unintended consequences, a dynamic interpretation of complexity prevails. These two facets of complexity, i.e. interdependency and emergence, and the non-linearity inherent in the underlying processes justifies taking a systems thinking approach than taking a reductionist approach in studying value co-creation in complex service

systems [1], [6].

The complexity in service systems leads to uncertainty in the environment of a given entity, wanting it to be viable in order to survive in such a turbulent environment. This idea is embedded in the Viable Systems Model (VSM) of Beer (1984) [14]. Based on the VSM as well as a rich stream of research in systems theories, [15] reports on Viable Systems Approach (VSA), which acts as a meta-model to interpret business and social phenomena. The goal of this line of research, according to [15], is to develop a methodological approach to address system dynamics in conditions of complexity to achieve system viability through a sustainable governance approach to business phenomena. The VSA proposes a model of firms (or any social entity) as viable systems, which enables their interpretation as open systems that aim to survive in their context by dynamically interacting with several other systems entities that the observed system perceives as relevant in that they offer resources that are critical for its functioning and viability.

2.2. *Value Co-creation*

Value, in the service science domain, is defined as a change that people prefer [13]. The Service-Dominant logic insists that value cannot be added beforehand to a product or a service and exchanged for something [16]. Instead, it is suggested that firms can only offer value propositions, along which customers could co-create value if they accept that value proposition. In other words, a value proposition is something through which a firm offers a service to its clients. According to [6], an increasing number of firms today consider their core proposition to be one of service even if they are still involved in manufacturing, not merely because they offer service activities “wrapped around” the equipment they manufacture, but also due to increasing recognition of the fact that service and manufacturing activities interact through the design, technology and business value propositions of their offerings. This leads to the notion of value-in-use, which is the value being realized at the time of use [16]. Moreover, a value proposition comprises a set of value creating activities / attributes, to which firms mobilize its resources and the amount of resources mobilized to each activity / attribute at a given time is known as density [16], [17].

The notion of value co-creation changes the role of the customer from a passive value recipient to an active co-creator. In other words, the customer is considered as an integral element of the service system, which contribute the resources accessible to them into the system, making their resources (e.g. knowing how to use an ATM or understanding how to get around an airport) part of organization’s service capability [14]. Furthermore, service is said to be often involving “value constellation”, which are networked and complex and which suggest a multi-faceted and iterative approach with the customer system within the whole system [14]. According to [18], value propositions establish connections and relationships among service systems and in value co-creation; value is ultimately derived with the participation of, and determined by, the beneficiary (often the customer) through use (often called ‘consumption’) in the process of acquisition, usage and disposal. Moreover, [18] defines value for service systems simply in terms of an improvement in system well being, which could be measured in terms of a system’s adaptiveness or ability to fit in its environment.

However, as mentioned before, the notion of value co-creation is highly conceptual and ambiguous [5], [6]. According to [17], despite discussions of and arguments for a service-dominant logic approach, there has been, to date, little work in

trying to operationalize the logic. As identified in [19], there are four phases within the process of value co-creation namely co-experience, co-definition, co-elevation and co-development. Co-experience is the sharing of internal models of preferences, capabilities and expectations of each other in a collaborative value co-creation process between customers and providers. Co-definition results in a shared common internal model by mutually learning each other. Co-elevation is a zigzag shaped spiral up process of expectation of the customers and abilities of the providers. Co-development pays attention to co-innovation generated by simultaneous collaboration among various entities. Here, co-experience and co-definition are said to be short-term whereas co-elevation and co-development are long-term. The two short-term sub processes of value co-creation, i.e. co-experience and co-definition have been modeled as a symbiotic hyper game in [20].

2.3. Agent-based Modeling

Agent-based modeling consist of a number of interacting autonomous agents who are represented as computerized independent entities capable of acting locally in response to stimuli or to communication from other agents [21]. These agents, when put together, interact with each other according to their local information and behavioral rules, resulting in various complex patterns. In other words, these agents act as parts of a complex whole, of which the properties can only be studied by letting the parts to interact with each other. Due to this reason, agent-based modeling has become a prominent technology in studying complex adaptive systems [22].

According to Allan Kay, the best way to predict the future is to invent it [22]. In social science, this is called the generative approach, in which a generativist looking forward to explain the emergence of macroscopic societal regularities, such as norms or price equilibrium, would like to know how the decentralized local interactions of heterogeneous autonomous agents could generate the given regularity [7]. Generally, the interdependency, emergence and non-linearity inherent in the underlying processes make it difficult for humans, unassisted by computer simulations, to effectively reason about the consequences of actions in a complex system [23]. Agent-based modeling enables to generate that future [22], i. e. the would be world [24], in the form of a computer simulation in which a group of heterogeneous, autonomous, bounded rational agents interact locally in a explicit space [7]. The creation of silicon surrogates of real-world complex systems allows us to perform controlled repeatable experiments on the real McCoy [24].

Even though agent-based modeling has been used extensively in various domains of complex adaptive systems, there is not much evidence in the literature about using agent-based approach to study complex service systems. However, the emerging literature in complex service systems implies that there is a need for adopting such computational techniques for the progress of the discipline [1], [15], [6], [18]. According to [26], one key challenge in developing a new science of services is in finding appropriate methods for modeling service systems. Meanwhile, [23] sees that agent-based modeling techniques, first developed for artificial intelligence, are now being applied in new areas such as computational organization theory and agent-based computational economics, indicating an interdisciplinary academic shift with a potential for services. Furthermore, [10] proposes agent-based modeling as a research methodology for this emerging field, which lacks research methodologies.

3. Kauffman's NKCS Model

The NKCS model developed by Kauffman mimics the co-evolution of multiple species in a biological ecosystem [11]. This can be likened to the process, in which stakeholders of a complex service system co-evolve outcomes, both individually as well as systemically, by providing service to each other by means of their individual competence. Thus, the NKCS model provides an appropriate and interesting basis for a rational discussion on value co-creation in complex service systems.

As the name denotes, the NKCS model is based on four main parameters N, K, C and S. Even though not mentioned, there exist two other parameters namely X and A. The NKCS model defines S number of species, each represented by a genotype comprising N number of genes. K defines the degree of inter-dependence of each gene within a genotype. In other words, each gene in a given genotype depends on K number of genes of the same genotype. C defines the degree of interdependence of each gene in a given genotype with genes of another genotype. In other words, each gene in a genotype depends on C number of genes of another genotype. Parameter X denotes the number of other species in the system that a given species interacts with. Therefore, each gene of a given genotype depends on C number of genes of each of its X interacting genotypes.

Each species has an individual fitness landscape defined by its genotype as an N-dimensional hypercube. Each point in this landscape has a coordinate written as a string of digits with base A, where A is a positive integer. For example, if N = 5 and A = 2, points in this 5-dimensional hypercube could be identified as 00000, 00001, 00010, etc. Each point $d = d_1, d_2, \dots, d_n$ in this N-dimensional hypercube has an associated fitness value f as defined by Eq. (1).

$$f(d) = \frac{1}{N} \sum_{i=1}^N f_i\{d_i, [d_{i_1}, \dots, d_{i_K}], [(d_{i_1}, \dots, d_{i_{1C}}), \dots, (d_{i_X}, \dots, d_{i_{XC}})]\} \quad (1)$$

Here, f_i is the fitness contribution of gene d_i at locus i . However, it depends on the gene d_i as well as the other genes that d_i is depending on. With parameter K, the gene d_i depends on K other genes denoted by $[d_{i_1} \dots d_{i_K}]$. Moreover, with parameters C and X, the gene d_i depends on C number of genes in each of X number of other genotypes denoted by $[(d_{i_1}, \dots, d_{i_{1C}}), \dots, (d_{i_X}, \dots, d_{i_{XC}})]$. Position values (i_1, \dots, i_K) and $[(i_1, \dots, i_{1C}), \dots, (i_X, \dots, i_{XC})]$ are determined randomly. The value of f_i is determined by a function, which is defined by Eq. (2).

$$f_i: \{0, \dots, A - 1\}^{K+XC+1} \rightarrow R \quad (2)$$

Here, R is drawn from a uniform distribution in the range (0, 1) to each of its $A^{(K+C+1)}$ inputs. The different fitness values associated with each point in a given entity's landscape entails a rugged terrain for the entity (say, a representative agent) to traverse, from valleys to peaks, looking for better fitness values. However, due to the dependency imposed by parameter C, a movement of one entity may deform the position of another (possibly many) affecting its fitness. This provokes reaction from the affected entities in return and, as the process continues, all entities in the system move to positions with better fitness values. This process is identified as co-evolution of species.

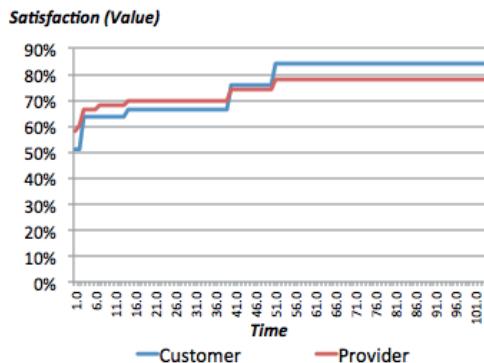


Figure 1. Co-evolution of satisfaction (Value) of the two agents, service provider and customer, with time

According to [11] there are three different strategies for the entities to use in traversing, namely one-mutant change, fitter dynamics and greedy dynamics. In one mutant change, a representative agent of an entity moves to a new location selected from its one-mutant neighbors. In the fitter dynamics strategy, the said agent moves to a one mutant location if the utility at that location looks greater than the current utility. In the greedy dynamics strategy, the said agent moves to the one-mutant neighbor with highest utility compared to the current utility.

4. Representing Value Co-creation – The Agent-based Example

Using the above NKCS framework as the basis, we formalize a simple agent based model to operationalize the process of value co-creation. Even though a typical complex service system may comprise several entities (species), for this discussion we use a model of two entities, service provider and customer, as the focal point is the representation of the value co-creation process. In our model, we have representative agents for each of the two entities, who are seeking for points with higher fitness values in their respective landscapes. We assume that the fitness of a particular point indicates a specific utility level of the respective agent. The landscapes are designed according to the NKCS architecture and we have set the parameters to $N = 5$, $K = 2$, $C = 2$. Since there are only two types of species, S becomes 2 and X becomes 1. Furthermore, we set $A = 2$.

The value, or to be more specific, the value in use, is defined in this system as agent's satisfaction. The agents follow the one-mutant change strategy to move from one point in their respective landscapes to another, to gain more satisfaction. We assume that the service provider agent's satisfaction is the average of his individual utility and the individual utility of the customer whereas the satisfaction of the customer is his individual utility itself. Thus;

$$S_p = (U_p + U_c)/2 \quad (3)$$

$$S_c = U_c \quad (4)$$

Here, S_p = Satisfaction of the provider, U_p = Individual utility of the provider, U_c = Individual utility of the customer and S_c = Satisfaction of the customer. Based on different types of providers and customers defined in [20], here we assume the service

provider agent to be ‘proactive’ and the customer agent to be just ‘taking’ the service as it is. Both agents are initially placed at random points in their respective landscapes. The service provider agent actively seeks for higher satisfaction by randomly moving to one of his one-mutant neighboring points [11] at each time step. If the satisfaction at that point is higher than the satisfaction at the previous point, he moves to the new point permanently. Otherwise, he steps back to the previous point. This simulation was conducted for 100 time steps and the Figure. 1 shows how the satisfaction of the customer and the service provider evolves with time.

5. Discussion

This section discusses how our simple agent-based model operationalizes value co-creation phenomenon and the implications for modeling complex service systems. We use a rather simple example to explain the basics and also introduce a published case of a real complex service system, to which our method could be applied. According to the literature on SD-Logic, firms offer value propositions, which comprise a set of value creating attributes, for customers to co-create value [16]. For example, a teacher’s (service provider) value proposition of teaching may comprise making lesson plans (*LP*), determining learning objectives (*DO*), doing reference (*DR*), making study materials (*SM*), using multimedia (*MM*), giving verbal instructions (*VI*) and evaluating continuously (*CE*). Similarly, customers also have a set of value creating attributes, which enable them to realize value of a given service at use. For example, a student’s (customer) value creating attributes in a service received from a teacher would be having clear objectives (*CO*), ability to memorize (*AM*), punctuality in attending classes (*PA*), time allocated to refer study materials (*TS*), developing necessary skills for learning (*DS*), enthusiasm in learning (*EL*) and effort put to obtain additional external knowledge (*EK*).

The multi-dimensional landscape structure of the NKCS architecture reasonably fits into this scenario. From Eq. (1) we can write,

$$f_{Teacher} = \frac{1}{7}(f_{LP} + f_{DO} + f_{DR} + f_{SM} + f_{MM} + f_{VI} + f_{CE}) \quad (5)$$

$$f_{Student} = \frac{1}{7}(f_{CO} + f_{AM} + f_{PA} + f_{TS} + f_{DS} + f_{EL} + f_{EK}) \quad (6)$$

A dimension (gene) of a landscape could be considered as a single value-creating attribute of a given species. The digit value of a gene at a given time could be considered as a measure of the level of resources that have been mobilized into that dimension at that particular time. For example, the number of hours that a teacher would spend to do reference about the respective subject would determine the resource level (digit value) of that particular value-creating attribute. Thus, a point of a species in the respective landscape would determine the density [16][17] of resources of the value proposition.

The value creating attributes are not independent in practice. They are rather interdependent. For example, we can assume that the student’s enthusiasm in learning (*EL*) would be enhanced by his / her ability to memorize (*AM*), and by having learning objectives (*LO*). Further we can assume that it requires the teacher to give effective verbal instructions (*VI*), use multimedia (*MM*) and prepare study materials (*SM*). This phenomenon is well represented in the NKCS architecture through the parameters K

and C. In this example, the attribute EL depends on two ($= K$) other attributes of the student him / herself and three ($= C$) attributes of the teacher. Hence, according to the Eq. (1), the utility (fitness) contribution of attribute EL can be written as $f_{EL}[EL, (AM, LO), (VI, MM, SM)]$. According to Eq. (2), value of f_{EL} can be drawn from $\{0, \dots, A - 1\}^{2+1*3+1} \rightarrow R$.

5.1. Operationalizing Value Co-creation

In the SSME literature on value co-creation, customer is considered as an integral element of the service system, which contributes the resources accessible to them into the system, making their resources part of service provider's service capability [14]. From $f_{EL}[EL, (AM, LO), (VI, MM, SM)]$, it is clear that the contribution of the attribute EL to the student's overall satisfaction depends on some attributes of the student him / her self (AM, LO) as well as on some attributes of the teacher (VI, MM, SM). In other words, a student without possessing a particular level of resources (digit value) in attributes AM and LO may not be able to co-create the best value even if the teacher possessing best levels of resources. This is true for the other side as well. The teacher's satisfaction depends on contributions from his / her individual value creating attributes and those contributions depends on resource levels (digit values) of all connected attributes of the teacher as well as the student. In other words, even a brilliant teacher may not be satisfied with teaching if his / her students do not possess necessary levels in their value creating attributes.

Given that value is an improvement of system well being [18], value co-creation should give rise to gradual improvement of satisfaction of both entities. According to [19], the provider's capabilities and the customer's desires grow up with time when they co-create value. This is clearly shown in Figure. 1, in which the satisfaction of both customer agent and service provider agent grows with time. It is noted that the system converges at $t = 50$, which we believe is due to the limitations imposed by the simplicity of this model. The model presented in this paper could be considered as a model of an active service provider and a passive customer. In other words, the service provider is actively seeking to improve the well being of the system whereas the customer is just waiting till the provider shows him the way. Hence, a passive learning environment of a classroom would be a typical example. However, according to [20], there are various types of customers as well as providers, whose different behaviors may change the resulting pattern of value co-creation in Figure. 1. Furthermore, the convergence at $t = 50$ could be considered as a point where a radical innovation or a paradigm shift is required. Computationally, this could be interpreted as the agents ending at local optima in their respective landscapes.

5.2. Implications for Modeling Complex Service Systems

According to [12], a complex service system is a value constellation. A value constellation comprises multiple entities connected via customer – service provider value co-creation interactions. This leads to the notion that every entity renders their competence to the system in the form of a service to co-create value. The case of London Borough of Sutton [25] provides a very good example of a complex service system. In this case, the Sutton Council and the Metropolitan Police in the London Borough of Sutton work closely together to improve community safety in the borough. As they have identified, the major issue that the residents of the borough are concerned

is the fear of crimes. Therefore, the two government entities regularly research and engage with residents and communities to reduce the fear of crimes in the borough. Authorities continuously investigate on the level of resources on various aspects of the communities and residents to improve their services. Communities and residents on the other hand contribute with their resources, such as time, ideas and information. The ultimate result is the less fear of crimes in residents, which is the value, co-created by them with the help of other stakeholder entities.

The example agent-based model presented in this paper obviously does not represent a complex service system. However, it could be easily extended to an agent-based model of a complex service system. The parameter S, which stands for the number of entities in the system, can be used to add more entities to the system. Parameter X could be used to represent an entity that is interacting with multiplicities of other entities. For example, in the case of the London Borough of Sutton, there are multiple entities such as the Sutton Council entity, the Police entity, Residents entity, etc. Furthermore, the Sutton Council entity maintains value co-creation relationships with both Police entity and the Residents entity. The value co-creation relationships between entities could be represented using the dyadic model presented in this paper. An agent-based model based on this structure may have multiplicities of agents belonging to each entity, each having different objectives, resource levels, utilities and behavioral rules. For example, the residents in the above example may comprise locals and immigrants, who possess completely different objectives and behaviors. Thus, agent-based modeling of complex service systems based on the proposed NKCS framework could be used as a means of studying value co-creation in complex service systems.

6. Conclusion

In this paper we presented a model of a value co-creation interaction between a service provider agent and a customer agent based on the Kauffman's NKCS architecture. The objective of this paper was to propose this method as a means to model the value co-creation process in agent-based models of complex service systems. We use basic concepts of value co-creation available in service science, management and engineering (SSME) literature and a toy example of a teacher –student interaction to discuss the applicability of the proposed method to represent value co-creation process. We further discuss the extendibility of the proposed method to represent value co-creation interactions in value constellations of complex service systems. Our ongoing research involves using the proposed approach to study complex service systems in the tourism domain.

References

- [1] Ng, L. Andreu, Special Issue: Research Perspectives in the Management of Complex Service Systems, *European Management Journal* 3(5), (2012), 405-409.
- [2] IfM and IBM (2008), *Succeeding through service innovation: A service perspective for education, research, business and government*, Cambridge, United Kingdom: University of Cambridge Institute for Manufacturing, 2008.
- [3] P. Maglio, J. Spohrer, Fundamentals of service science, *Journal of the Academy of Marketing Science*, 36, (2008), 18-20.

- [4] R. Lusch, S. Vargo, A. Malter, Marketing as Service Exchange: Taking a Leadership Role in Global Marketing Management, *Organizational Dynamics* 35, (2006), 264-278.
- [5] C. Grönroos, Value Co-creation in Service Logic - A Critical Analysis, *Marketing Theory* 11(3), (2011), 279-301.
- [6] R. Badinelli, S. Barile, I. Ng, F. Polese, M. Saviano, P. Di Nauta, Viable service systems and decision making in service management, *Journal of Service Management* 23(4), (2012), 498-526.
- [7] J. M. Epstein, Agent-based computational models and generative social science, Generative Social Science: *Studies in Agent-based Computational Modeling*, Joshua M. Epstein (Eds.), Princeton University Press, Princeton, New Jersey, 2006.
- [8] J. L. Casti, The Computer as a Laboratory: Towards a theory of complex, adaptive systems, *COMPLEXITY* 4(5), (1999), 12-14.
- [9] K. M. Carley, On Generating Hypothesis Using Computer Simulations, *DTIC Online: Information for the Defence Community*, 1999.
- [10] D. S. Utomo, U. S. Putro, P. Hermawan, Agent-based Research Methodology for Service Science, Management and Engineering (SSME) in Industrial Cluster, *2nd International Research Symposium in Service Management*, Yogyakarta, Indonesia, 2011.
- [11] R. Vidgen, J. Padgett, Sendero: An Extended, Agent-based Implementation of Kauffman's NKCS Model, *Journal of Artificial Societies and Social Simulation* 12(3), (2009)
- [12] C. A. Kliliszewski, P. P. Maglio, M. Cefkin, On modeling value constellations to understand complex service system interactions, *European Management Journal* 30(5), (2012), 438-450.
- [13] J. Spohrer, P. Maglio, Service Science: Toward a Smarter Planet, *Service Engineering*, Karwowski and Salvendy (Eds.) Wiley, New York, NY, 2009.
- [14] I. Ng, R. Maull, L. Smith, Embedding the new Discipline of Service Science, H. Demirkiran et al. (Eds.), *The Science of Service Systems, Service Science: Research and Innovations (SSRI) in the Service Economy*, Springer, 2011.
- [15] S. Barile, J. Pels, F. Polese, M. Saviano, An Introduction to the Viable Systems Approach and its Contribution to Marketing, *Journal of Business Marketing Management* 5(2), (2012), 54-78.
- [16] I. Ng, G. Parry, R. Maull, G. Briscoe, Transitioning from a Goods-Dominant logic to a Service-Dominant logic: Visualizing the Value Proposition of Rolls Royce, *Journal of Service Management* 23(3), (2012) 416-433.
- [17] S. Michel, S. Vargo, R. Lusch, Reconfiguration of the conceptual landscape: a tribute to the service logic of Richard Normann, *Journal of the Academy of Marketing Science* 36, (2008), 152-155.
- [18] S. L. Vargo, P. P. Maglio, M. Archpru Akaka, On value and value co-creation: A service systems and service logic perspective, *European Management Journal* 26, (2008), 145-152.
- [19] K. Kijima, Value Orchestration Platform for Value Co-creation, *Frontier of Service System Science*, Tokyo, 2012 (presentation).
- [20] S. Novani, K. Kijima, Symbiotic Hypergame Analysis of Value Co-creation Process in Service Systems, In Proc. 7th International Conference on Service Systems and Service Management "ICSSSM", 2010.
- [21] M. A. Janssen, Complexity and Ecosystem Management: *The Theory and Practice of Multi-agent Systems*, Marco A. Janssen (Eds.), Edward Elgar Publishing Inc., Cheltenham, UK, ISBN: 1 84376 0614, 2000.
- [22] T. Terano, Beyond the KISS Principle of Agent-based Social Simulation, *Journal of Socio-informatics* 1(1), (2008), 175-187.
- [23] J. Spohrer, P. Maglio, The Emergence of Service Science: Toward Systematic Service Innovations to Accelerate Co-creation of Value, *Production and Operations Management* 17, (2008), 1-9.
- [24] J. L. Casti, Would-be Worlds: How Simulation Is Changing The Frontiers Of Science, John Wiley and Sons, Inc., 1997.
- [25] L. Andreu, I. Ng, R. Maull, W. Shadbolt, Reducing the fear of crime in a community as a complex service system: The case of London Borough of Sutton, *European Management Journal* 30(5), (2012), 410-417.
- [26] P. Maglio, Modeling Complex Service Systems, *Service Science* 3(4), (2011), i-ii (Editorial).

Agent-Based Simulator for Travelers Multimodal Mobility

Mahdi ZARGAYOUNA^{a,1}, Besma ZEDDINI^a, Gérard SCEMAMA^a and Amine OTHMAN^a

^a Université Paris-Est, IFSTTAR, GRETTIA, France

Abstract. Transport systems are more and more complex and have to evolve to integrate more connected entities (mobile devices, localized vehicles, etc.). It becomes critical to develop micro-simulation tools for mobility policies makers taking into account this fact. In this paper, we propose a multimodal travel simulator that allows for the understanding and the prediction of future status of the networks and allows for testing new online applications. The application simulates the movements of travelers on the different networks while taking into account the changes in travel times and the status of the networks. The considered transport modes include pedestrians, private cars, all public transport modes and ridesharing. The simulator has been developed using the Repast Simphony® multiagent simulation platform.

Keywords. Transportation, Multimodal services, Modeling, Simulation.

1. Introduction

Multimodal traffic information and control platforms such as [1] integrate different information sources about different transport modes. They allow for a common visualization of the different networks states, and the followup of several indicators (regularity, advance, delay, etc.) related to the different networks components (public transport lines, itineraries, roads, etc.). However, current platforms provides rarely integrated simulation tools that would predict the effect of control actions on the multimodal networks. More generally, transport systems are becoming more and more complex and have to evolve to integrate more connected entities (mobile devices, localized vehicles, trackable goods, etc.), and it becomes critical to develop micro-simulation tools for mobility policies makers capable of representing and managing these entities.

In this paper, we propose a multimodal travel simulator that allows for the understanding and the prediction of future status of the networks and that can be used for testing new online applications that track individual travelers. Online applications are applications that track individual travelers and need to be con-

¹Corresponding Author: Mahdi Zargayouna, Université Paris-Est, IFSTTAR, GRETTIA, Boulevard Newton, Champs sur Marne F-77447 Marne la Vallée Cedex 2, France; E-mail: hamza-mahdi.zargayouna@ifsttar.fr.

tinuously aware of their positions. Current tested applications are urban parking, dial-a-ride and crisis management. The application simulates the movements of travelers on the different networks while taking into account the changes in travel times and the status of the networks. In this paper, we present the main building blocks of the simulator, which are easily reproducible in other simulators pursuing similar objectives.

The multiagent paradigm is relevant for the simulation of urban transport systems. It indeed facilitates an approach by analogy in the transport domain which one of the objectives is the coordination of distributed entities. This is why the multiagent approach is often chosen to model, solve and simulate transport problems [2]. This approach is particularly relevant for the simulation of travelers mobility since the objective is to take into account human behaviors that interact in a complex, dynamic and open environment. The proposed simulator is composed of autonomous agents, evolving in an environment of which they have a partial perception.

The remainder of this paper is structured as follows. In section 2, we discuss the choice of the simulation platform and previous proposals for travelers mobility simulation. Section 3 presents the parameters and data of the simulator. Section 4 describes the behaviors of the agents in our system. In the section 5, we detail the multimodal path planning and the representation of the multimodal networks before to conclude and describe some further work we are conducting.

2. Related Work

To design and implement a multiagent simulator, it is possible to develop an application directly in a host programming language. However, it is often faster, more useful and more efficient to ground the simulator on an existent multiagent simulation platform. In the context of transportation applications, one main choice criterion for the simulation platform is its ability to create geospatial agent-based models, i.e. its ability to integrate and process geographic data. Based on this criterion, we haven't considered several popular multiagent platforms such as Jade [3], Mason [4] and Madkit [5], for which it is difficult to integrate GIS capabilities.

Swarm [6] is a simulation platform that has a library allowing for the loading of layers of GIS data. However, it does not provide spatial primitives nor gives the possibility to store the resulted environment [7]. NetLogo allows to import and export GIS data and provides some basic geometrical operations, but not the more advanced spatial analysis operations. Gama [7] provides an environment for building spatially explicit agent-based simulations, while Repast Simphony [8] integrates a GIS library (Geotools), and provides additional GIS services (network modeling as a graph, computation of shortest paths, visualization and management of 2D and 3D data, etc.). We have chosen the Repast Simphony platform, since between all the available simulation platforms that fit with our requirements, it is the most mature one and more importantly, it integrates several API to deploy simulators over several hosts².

²The distribution of our simulator over several hosts is one of our ongoing works.

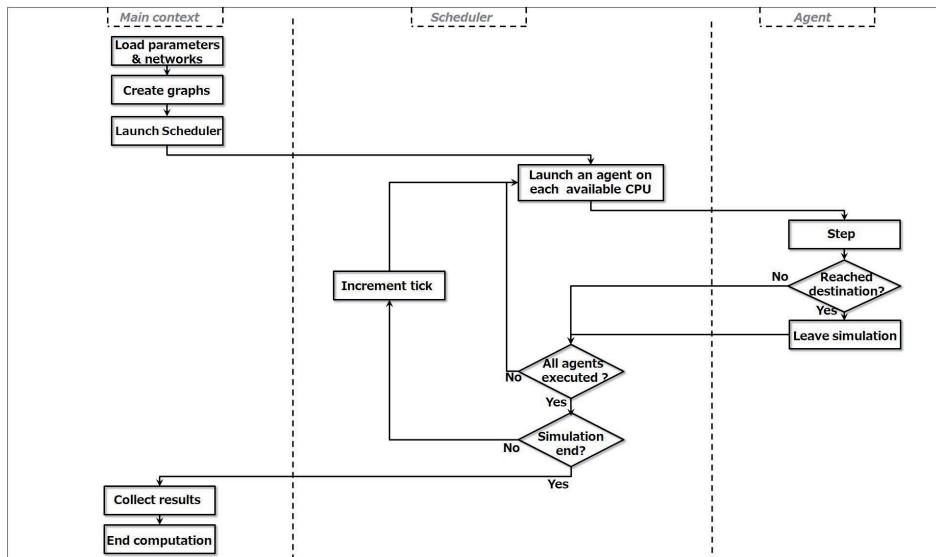


Figure 1. Workflow of the simulation

Besides, there exists several multiagent simulators for travelers mobility. For instance, Transims [9] simulates multimodal movements and evaluates impacts of policy changes in traffic or demographic characteristics. Miro [10] reproduces the urban dynamics of a French city and proposes a prototype of multiagent simulation that is able to test planning scenarios and to specify individuals' behaviors. However, none of these proposals assume the continuous localization of travelers and means of transport. In addition, none of them integrate dynamic ridesharing as one of the transport modes in the multimodal network. In the proposal described in this paper, travels are multimodal: they concern private cars, all the public transport modes as well as ridesharing and pedestrians. Passengers routes are monitored and alternatives are proposed to them if something wrong happens with their itinerary.

3. Data and Parameters

The workflow of Figure 1 details one simulation execution and structures the remainder of the presentation. The simulation starts with the loading of the parameters (simulation duration, number of agents of each type, the default speeds of each agent type, etc.). Then, the main context program creates the logical graphs as described in section 5 and launches the scheduler. Reapst is based on a discrete event scheduler. The default scheduler iterates over the agents and executes their *step* method. We have replaced it by a new scheduler that launches the active agents in parallel over all the available CPUs. When launched, each agent executes his *step* method. The behavior of each agent type is described in section 4. Finally, when the simulation duration is reached, the results are reported and the simulation ends.

3.1. Data

The input data of the simulator are: i) the road network, ii) the public transport network, iii) the transfer mapping, iv) the timetables of the public transport vehicles, v) the pedestrian network, vi) the travel patterns, vii) the travelers profiles. The road network is a description of the roads, crossroads and driving directions. Each road has, among other information, a corresponding minimum and maximum speeds. It also has a mapping between traffic flux (vehicles/hour), the traffic density and speeds. A public transport network is composed of transport lines, each of which composed of a set of itineraries. An itinerary is composed of a sequence of edges. Each edge has a tracing in the form of a sequence of pairs $\langle longitude, latitude \rangle$, and is composed of an origin node and a destination node. Finally, every node is defined by its name and coordinates. The transfer mapping is a table informing about the stops of the network for which a transfer by foot is possible and the road transport nodes that are reachable from the stops. The timetables of the vehicles are composed of a set of missions. Each mission corresponds to a specific itinerary³ and describes the path of a vehicle and the corresponding visit times. Each timetable is then a sequence of pairs $\langle stop, time \rangle$. The pedestrian network is a subset of the road network in which pedestrians can move, but which is undirected (pedestrians don't have to obey to the one-way limitations). A travel pattern clusters the region in zones and describes the number of persons asking to leave or to join each region. The travelers profiles define the properties and preferences of the travelers. Most importantly for the simulator, they define which drivers and which passenger are interested by the ridesharing service. The preferences of the traveler also define the accepted time gap between their computed itinerary and their real situation, before asking for a new up-to-date itinerary.

3.2. Parameters

3.2.1. Simulation duration

Two values define the duration of a simulation run. The first is an interval defining the first and last date that are simulated (noted τ^- and τ^+ respectively). In the absence of these parameters, we use respectively the values associated to the maximum date and minimum date in the timetables of the public transport vehicles. The second value that has to be defined is the number of discrete ticks of time that the simulation will execute before terminating (noted δ henceforth). At each tick, all the agents are activated for a particular action defined by their behavior.

3.2.2. Agents speeds

The speeds of the agents are defined in three ways:

- Public transport vehicles: based on the input data, they are inferred from the $\langle stop, time \rangle$ pairs.

³This correspondance mission-itinerary has to be kept in mind as we will be referring to it in section 5.

- Private cars: based on the maps data, their speeds are taken from the mapping density/speed of the road they are currently traversing.
- Default: the simulator user defines speeds for pedestrian, cars and public transport vehicles as a parameter. The user-defined mean car speed is used if the speed data is missing from the current road of the car. The user-defined public transport speed is used if two successor pairs $\langle stop, time \rangle$ give an inconsistent speed due to errors in the data⁴.

3.2.3. Units transformation

All the data that are expressed in function of time (e.g. the visit times of the public transport vehicles) have to be expressed in terms of simulation ticks. However, since the original time data are expressed in terms of date, they have to be transformed as follows. Let t the original time, t' is the new time (in simulation tick), computed this way:

$$t' = \frac{t - \tau^-}{\gamma} \times \delta, \text{with } \gamma = \tau^+ - \tau^-$$

In addition, all the speeds are originally defined in terms of Km/h. They have to be transformed into meters/tick as follows (σ_{mode} is the speed of the transport mode, and $\sigma_{modeKmH}$ is its original speed expressed in Km/h).

$$\sigma_{mode} = \frac{\sigma_{modeKmH} \times \gamma}{3.6 \times \delta}$$

Let us take the private car mode as an example. $\frac{\gamma}{\delta}$ gives the number of seconds elapsed in a simulation tick of time. Thus, $\sigma_{car} = \frac{\sigma_{carKmH} \times \gamma}{3.6 \times \delta}$ gives the number of meters that the car can travel in a tick of simulation time. The same principle applies for public transport vehicles and pedestrians.

4. Multiagent System

4.1. Planner Agents

The planner agents have the responsibility of computing the best road itinerary for the car agents and the best multimodal itinerary for the traveler agents. A planner agent is created when an agent request is submitted to the system, and leaves the system right after. Planner agents base their computation on the latest status of the networks. The graph representation and the paths calculations by the planner agents are described in section 5. A road plan is composed of a sequence of edges together with their corresponding visit times. A multimodal itinerary is composed of a sequence of pairs $\langle id_{vehicle}, itinerary \rangle$, with $id_{vehicle}$ the identifier of the vehicle to take and the corresponding part of the itinerary.

⁴E.g. infinite speed due to two identical visit times of two successive stops.

4.2. Agents Movements

Each active agent has a list of coordinates that he has to follow, resulting from the itinerary that he received from a planner agent. Agents are allowed to move for a certain distance at each tick equal to σ_{car} , $\sigma_{PTvehicle}$ or $\sigma_{passenger}$ depending on the type of agent. At each tick, the agent iteratively checks if he can move from his current coordinate to the next one in his list. If not, he calculates the intermediate coordinate that corresponds with the remaining distance that he is allowed to travel. In the next tick, the agent can travel the remaining distance to the next coordinate.

4.3. Car Agents

When created, a car agent has an origin and a destination. If travel patterns exist for the considered network (cf. section 3), the origin and destination of each car agent are chosen such as the origins and destinations of all agents are proportional to the pattern. If not, the origin and destination of the car agent are chosen randomly. He then asks the planner for the best itinerary between his origin and his destination. If the driver profile states that the corresponding car agent is interested in ridesharing, the car agent registers himself in the nodes of his itinerary (cf. section 5 for the use of these registrations). At each simulation tick, the car agent checks if he has reached his destination. If so, he leaves the simulation. If not, he keeps on travelling.

4.4. Public Transport Vehicle Agents

The public transport vehicle agents don't choose their origin and destination and obey to predefined timetables. Each vehicle agent when created, infers his itinerary from his timetable. While the vehicle agent has not reached his destination, he travels at each tick the allowed distance, following his current speed. All the onboard passengers are moved to the same coordinates at the same time by the vehicle. That means that, when they are onboard a vehicle or a car, traveler agents delegate the control of their movements to the vehicle agent when they decide to take him. When the vehicle reaches a stop, he looks in his onboard travelers who has to leave at this stop. Then he looks in the waiting travelers at the stop who has to take him.

4.5. Traveler Agents

As for car agents, the origin and destination of the traveler agent are either inferred from travel patterns if they exist, or are chosen randomly. When they are not walking, traveler agents do not travel on their own, but share rides with other drivers and/or take public transport vehicles, which are responsible of their movements. The traveler agent alternates between walking and waiting for a vehicle (car or public transport).

4.6. Itinerary Monitoring

Each traveler agent and each car agent monitor their itinerary. The monitoring purpose is to ask for a new itinerary in two cases: a) the real position of the agent is different from the planned one with a certain gap Δ_a (defined in the preferences of the agent); b) there is an event on the transport network that impacts the traveler or the car planned itinerary. To be aware of the only events that concern him, the agent subscribes to the only edges of the transport network that form his itinerary. When the travel time of an edge changes, the new travel time is broadcasted to the subscribed agents. The planning process described in the previous paragraph is then launched with the new travel times associated with the network.

5. Multimodal Itinerary Planning

The planner agent is responsible for computing the itineraries for cars and passengers. To this end, he executes the A-Star algorithm [11] on the road and multimodal networks. The value of h - heuristic - associated to each vertex v for a specific destination v_d is equal to the geodesic distance between v and v_d divided by the maximum speed of the vehicles in the system. Recall that for a heuristic to be valid in A-Star, it doesn't have to be overestimated, which is verified for the values that we have chosen. However, for passengers, the computed itinerary could be the shortest in terms of theoretical travel time, but would result in too many transfers if the computed path contains too many different itineraries. Solving this problem is the purpose of the next subsection.

5.1. Public Transport Itinerary Planning

In order to avoid having passengers' shortest paths that pass by too many itineraries and thus result in too many transfers, we modify the public transport graph as follows (cf. Figure 2). Let $\langle s_1, s_2 \rangle$ an edge in the public transport graph. Let it_1 and it_2 two itineraries passing by this edge. We create four new vertices s'_1 , s'_2 that we connect with the edge that belongs to it_1 and s_1'' and s_2'' that we connect with the edge that belongs to it_2 . We also create four new edges $\langle s'_1, s_1'' \rangle$, $\langle s_1'', s'_1 \rangle$, $\langle s'_2, s_2'' \rangle$ and $\langle s_2'', s'_2 \rangle$, that we note pedestrian edges and that are heavily penalized. When we run a shortest path algorithm on this new graph, the itineraries that are on the same vehicle/itinerary are encouraged and a transfer is only proposed when it is impossible or really expensive to stay on the same vehicle. The transfer map is modified so that the stops or the crossroads that are reachable from s_1 (s_A in the figure) by foot become reachable from s'_1 and s_1'' (the red dotted circle in the figure). After applying the A-Star shortest path algorithm on the resulting itinerary, the planner agent interrogates the stops of the best found itinerary to infer the sequence of vehicles that the passenger agent will have to take and sends back the result to the traveler agent.

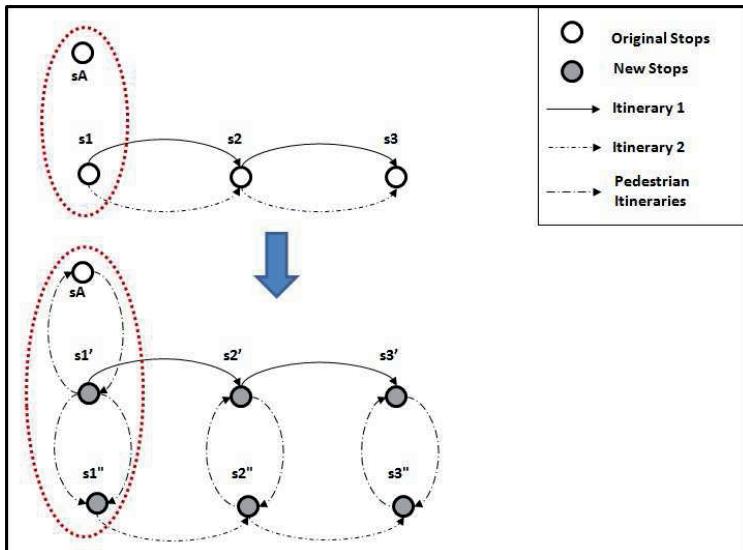


Figure 2. Multimodal Network Transformation

5.2. Ridesharing

In this paper, we consider a ridesharing service that is designed as a complementary service to public transport. We believe that this increases the chances for ridesharing to be a successful service. We consider that passenger would hardly accept to take more than one private car in a single itinerary. Based on these assumptions, if the passenger preferences indicate that he would be interested in ridesharing, the planner agent will look for these types of itineraries, following this order of priority:

1. Find a ridesharing itinerary with a single vehicle from the origin to the destination
2. Find a ridesharing itinerary from the origin followed by a public transport itinerary, or a public transport itinerary followed by a ridesharing itinerary to the destination
3. Find a public transport itinerary

Let C_o and C_d the cars that are registered in node o and node d (resp. the closest node to the origin of the traveler and the closest node to the destination of the traveler). To see if there is a ridesharing itinerary with a single vehicle from the origin to the destination, we calculate $C = C_o \cap C_d$. If $C \neq \emptyset$, the car $c \in C$ that respects the time constraints of the traveler and that arrive the soonest to d is chosen and his itinerary sent to the traveler agent. If no such vehicle is found, i.e. no car can transport the traveler from his origin to his destination directly, we look for itineraries which start or finish by a public transport itinerary. To this end, let $it_t = v_1, \dots, v_n$ the itinerary of the traveler agent. The planner agent computes iteratively the intersection of the vehicles registered in the successive nodes starting from v_1 looking for the vehicle that takes the traveler the furthest

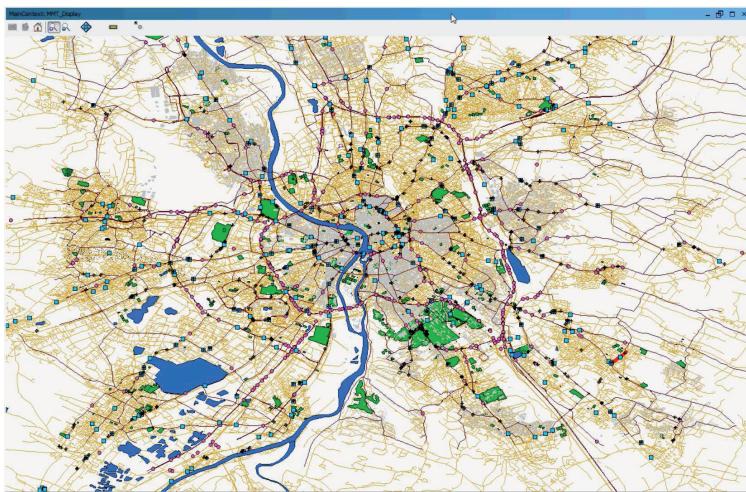


Figure 3. Simulation Execution

while respecting his time constraints. The same process is repeated starting from v_n and backwards looking for the vehicle that can transport the traveler to his destination. If such vehicle is found, the remainder of the itinerary is computed on the public transport network as explained in the previous paragraph. Finally, if none of these calculations succeeds, the shortest path in public transport is calculated.

We demonstrate the use of the platform for the city of Toulouse (cf. Figure 3), for which we have detailed data, including urban travel demand models. We have considered the main roads of the transport network of Toulouse with 13,226 roads. We also have considered the public transport network of Toulouse, with 80 lines, 359 itineraries and 3,887 arcs. In our current simulations, our multi-agent system is made of 118,270 agents: 28,720 buses⁵, 30,000 cars, 30,000 drivers and 30,000 passengers. Travel times are updated with real data from the ClaireSiti platform [1].

Since the transport network parameters feed our simulator, the positions of our agents are compliant with the real life situation. In the future, we plan to implement two identical applications with two different simulators that we would compare quantitatively.

6. Conclusion and Perspectives

In this paper, we have described the main building blocks of an agent-based simulator for multimodal travelers, which are easily reproducible in other simulators pursuing similar objectives. We have focused on the main capabilities of the agents together with the data and parameters necessary for such a simulator to work properly. This application simulates individual spatiotemporal positions that are

⁵In our simulations, we create one bus per mission.

compatible with the real situation. It is now possible to integrate transport applications that use these positions to provide an advanced service, such as route guidance, urban parking assistance, etc.

In the near future, we plan to develop scenarios with hundreds of thousands of passengers and drivers to verify that our implementation is actually scalable. Our second ongoing research is related to the first one. When we simulate realistic number of agents in the transport network, the problem of equilibrium arises. Indeed, if we send too many cars or travelers to the same itineraries, we risk to create congestions ourselves. We has to assign travelers and cars to itineraries while respecting their distribution aver competitive paths and not send them all to the same roads or vehicles.

References

- [1] G. Scemama and O. Carles. Claire-siti, public road transport network management control: a unified approach. In *Proc. of 12th IEE Int. Conf. on Road Transport Information & Control (RTIC'04)*, pages 11–18, London (UK), April 2004.
- [2] P. Davidsson, L. Henesey, L. Ramstedt, J. Tornquist, and F. Wernstedt. An analysis of agent-based approaches to transport logistics. *Transportation Research Part C - Emerging Technologies*, 13(4):255–271, 2005.
- [3] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.
- [4] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, July 2005.
- [5] O. Gutknecht and J. Ferber. The madkit agent platform architecture. In *Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55, London, UK, UK, 2001. Springer-Verlag.
- [6] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system: A toolkit for building multi-agent simulations. Technical report, Santa Fe Institute, 1996.
- [7] Patrick Taillandier, Duc-An Vo, Edouard Amouroux, and Alexis Drogoul. Gama: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *PRIMA*, volume 7057 of *Lecture Notes in Computer Science*, pages 242–258. Springer, 2012.
- [8] E. Tatara and J. Ozik. How to build an agent-based model iii – repast simphony. In *Applied Agent-based Modeling in Management Research, Academy of Management Annual Meeting, Chicago*, 2009.
- [9] K. Nagel and M. Rickert. Parallel implementation of the transims micro-simulation. *Parallel Computing*, 27(12):1611–1639, 2001.
- [10] S. Chipeaux, F. Bouquet, C. Lang, and N. Marilleau. Modelling of complex systems with AML as realized in MIRO project. In *LAFLang 2011 workshop*, pages 159–162, Lyon, France, 2011. IEEE Computer Society.
- [11] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.

Multi-Level Agent-Based Modeling: a Generic Approach and an Implementation

Đức-Ân VÕ^{a, b, 1}, Alexis DROGOUL^{a, c, d} and Jean-Daniel ZUCKER^{a, b, c}

^aIRD, UMI 209 UMMISCO, 32 avenue Henri Varagnat, Bondy, France

^bIFI, MSI, Vietnam National University, 42 Ta Quang Buu, Hanoi, Vietnam

^cUPMC Université Pierre et Marie Curie, UMI 209 UMMISCO, F-75005, Paris, France

^dDREAM, Can Tho University, 3/2 street, Ninh Kieu district, Can Tho, Vietnam

Abstract. Multi-level agent-based modeling (ML-ABM) requires representing agents at different levels of representation in the same model w.r.t. to time, space and behavior. This paper describes a generic and operational proposal for ML-ABM. First, a generic meta-model for ML-ABM and an associated “morphogenesis” operation are introduced. The generic meta-model allows a modeler to describe multiple levels of representation in the same model, while the “morphogenesis” operation supports agent change of representation level dynamically during the course of the simulation. Second, in order to demonstrate how to operationalize the proposal, we present an implementation of the generic meta-model and the “morphogenesis” operation in the GAMA ABM platform. Finally, we illustrate how our proposal, implemented in the GAMA platform, allows modeler in practice to develop a multi-level agent-based model. To do so, we rely on the famous “Boids” model of Craig Reynolds and show how the modeler can easily introduce a new level of representation of an entity to transform the model into a two-levels agent-based model without having to modify the existing model.

Keywords. Multi-level agent-based modeling, simulation, abstraction, meta-model, multi-scale, GAMA

1. Introduction

Agent-based modeling (ABM) has proved to be a successful paradigm for complex system studying for more than two decades [1]. This modeling approach can be viewed as a successful crossbreeding of Individual-Based Modeling (IBM), originated in Ecology, and Object-Oriented Programming (OOP), originated in Computer Science. Like the former, it aims at building generative models where the general behavior of the model is the result of the interactions between its components; like the latter, it allows building these models in a modular and incremental way, promising reusability and flexibility to the modelers. However, it also inherits some of the limits of its two ancestors, in particular the absence of dedicated abstractions for handling *more than one level of representation* simultaneously in a same model [2].

Recent years, researchers of the community have begun to propose solutions to facilitate the development of multi-level agent-based model. But existing approaches

¹ Corresponding Author: Đức-Ân VÕ, Associate Researcher, UMI 209 UMMISCO/IRD, 42 Ta Quang Buu, Hai Ba Trung district, Hanoi, Vietnam; E-mail: voducanvn@yahoo.com

still suffer from two significant drawbacks, which constitute obstacles to their adoption by the modelers. These approaches are either domain specific or only conceptual proposals. While current agent-based platforms still lack of appropriate abstractions, which explicitly support ML-ABM. We describe in this article a generic proposal and its implementation in the GAMA platform, which offers the modeler a fully operational solution for multi-level agent-based model development.

The organization of this article is as follows: the second section presents the modeler's requirements regarding multi-level representation in ABM using the Brown Plant Hopper (BPH) model. Then, in the third section, a quick review of current works dedicated to ML-ABM is given. The fourth section presents our proposal of a generic meta-model for ML-ABM and an operation for dynamically changing the representation level. In the fifth section, an implementation of our proposal in the GAMA platform is described. The sixth section illustrates how our proposal, implemented in the GAMA platform, makes the development of a two-levels agent-based model by a modeler easier. Finally, the last section concludes and gives some research perspectives of the work.

2. Modeler's Requirements for Multi-level Agent-based Modeling

Our proposal has been formalized after a careful analysis of the requirements of modelers working on several multi-level agent-based models and a specific attention to providing them with reusable and generic abstractions. To illustrate these modelers' requirements, we choose to present in the section the BPH model. This model aims at studying the effectiveness of stakeholders' decisions against the invasions of the BPH in the Mekong delta region. The modeler's requirements of this model cover the requirements of existing multi-level agent-based models in the community.

Besides being the basic daily food, rice plays an important role in the economy of Vietnam with more than 7.1 million tons export and a revenue of more than 3.5 billions USD in 2011 [3]. BPH, a kind of Nilaparvata species, is known as the major pest on rice and as the carriers of rice viruses-related diseases such as grassy stunt, ragged stunt and wilted stunt. The agricultural ministry of Vietnam reported a loss of 7000 tons of rice in 2006 due to the BPH invasions. To cope with the recurrent invasions of BPH, different control techniques and policies at various spatial and temporal scales have been applied. They range from the use of pesticides to synchronous cropping, development of genetically modified varieties, biological management. [4]

In order to assess the effectiveness and impact of control policies, since 2009, we have collaborated with the university of Can Tho and the Southern Regional Plant Protection Center to develop an integrated agent-based model. Developing such model implies the requirement to take into account multiple entities with their own policies, goals and scales of intervention (see Figure 1).

In summary, entities of this model possess three following characteristics. The first characteristic is that "entities" concerned with the control policies, e.g., decision-makers at different management levels, farmers, researcher, rural planer, companies, define different levels of organization related to specific spatial and temporal scales. For example, a decision-maker at the province level, e.g., a person in charge of controlling the BPH invasion of a province, issues policies that influence a province (spatial scale) within a period of several years (temporal scale). Figure 2 illustrates this

characteristic. Entities concerned with different control policies, i.e., decisional levels, are related to different spatial and temporal scales.

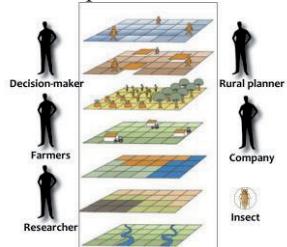


Figure 1. Entities with different policies, goals and scales of intervention in the BPH model.

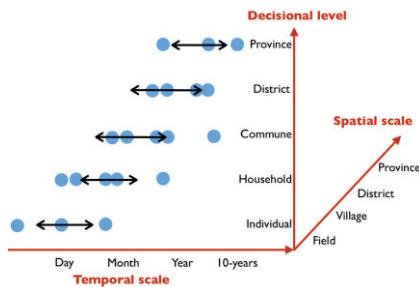


Figure 2. The relation of entities at different decisional levels with spatial and temporal scales.

The second characteristic is that some of these levels are hierarchical and there are privileged interactions between them. There are two types of hierarchies in this model. The first type of hierarchy is the decisional hierarchy. Stakeholders at different management levels, e.g., province, district, commune, are organized as the decisional hierarchy. There are privileged interactions and data exchanges between entities in the decisional hierarchy. For example, a stakeholder at the province level needs to interact and exchange data (e.g., reports of the BPH density) with several stakeholders at the underlying districts in order to issue control policies. The second type of hierarchy is the constitutive hierarchy in which an entity is considered as a member of another entity. For example, if the modeler would like to consider the BPH dynamics at three levels of granularity: individual insect, group of insect (moving between rice fields) and cloud of insect (migrating in the sky) then he needs to represent that a “cloud” entity contains groups of insects as members which in turn contain individual insects as members.

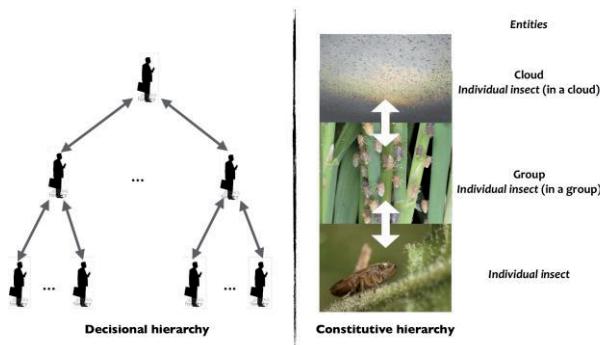


Figure 3. Hierarchies and multiple levels of representation of an entity.

The third characteristic is that some “entities” may need to be represented at different levels of organization in the same model. The level at which an entity is represented can dynamically change, depending on the presence or absence of other levels in the model. The modeler needs the possibility to dynamically switch between different levels of representation of an entity. For example, the BPH can be represented simultaneously at three levels: a separate individual insect, an individual insect in a group (of BPH moving in rice field), an individual insect in a cloud (of BPH migrating in the sky). The reason to mutually keep several representations of an entity in a model may vary. For example, an insect is considered as a separate individual if the modeler needs to take into account the detail life cycle of BPH. If the modeler wishes to take into account the movements of groups of insects within rice fields, he may need to represent individual insects as members of these groups. If the modeler wishes to consider the migration of insects in the sky, he may need to consider individual insects in clouds. The presence of “group” and “cloud” introduce two new representation levels of BPH, respectively “insect in group” and “insect in cloud”.

As mentioned in the beginning of this section, the modeler’s requirements of BPH model cover the modelers’ requirements of existing multi-level agent-based models that we have analyzed. We may generalize these characteristics of the BPH models as three modeler’s requirements of multi-level agent-based modeling (R1, R2, R3):

- R1. Entities may define different *levels of organization* related to specific spatial and temporal scales.
- R2. These levels of organization may be hierarchical.
- R3. During the simulation, some “entities” may need to be represented at different levels of organization.

3. State of the Art

Current approaches to ML-ABM suffer from two major drawbacks. The first one is that they are often domain specific, which requires the development of dedicated simulators such as RIVAGE, SimulBogota [5, 6]. These implementations raise the problem of reusability and generality. The second one is that many of the approaches are conceptual offering formal meta-models but no operational implementation in agent-based modeling platform (See for example [7, 8]). These latter approaches require the modeler to re-implement the formal meta-model from scratch, which is an extremely difficult task especially for non-programmer modelers.

Current agent-based modeling platforms (such as Repast [9] and NetLogo [10]) do not offer appropriate abstractions to support ML-ABM. Figure 4 represents a diagram with three levels: “Meta-model”, “Model” and “Simulation”. The “Meta-model” part describes the meta-model of current agent-based modeling platform. It offers concepts to describe an agent-based model. The “Model” part gives an example of an agent-based model specified using the concepts of meta-model. The “Simulation” part shows an example of a simulation initialized from the model.

Basically, current agent-based modeling platforms rely on a meta-model with three principal concepts: *Agent*, *Environment* and *Scheduler*. Agent represents the concept of agent in the model, which also defines a level of organization. Environment supports modeling the environment in which agents are situated, i.e., the spatial scale. Scheduler schedules the execution of agents during the simulation, defining the temporal scale.

Such meta-model fails to answer three requirements of the modeler summarized in the end of the previous section. The first requirement is that different agents can belong to different spatial and temporal scales. This meta-model implies that all agents in a model share a unique spatial scale (Environment) and a unique temporal scale (all agents are scheduled by the modeler in each simulation step). The second requirement raises the necessity to organize agents into hierarchy. Such hierarchy eases the modeling of interactions and data exchanges between agents of different levels. In order to facilitate the representation of hierarchy (of agents), the meta-model needs to natively support the concept of recursive agent. A recursive agent is an agent that can contain other agents as members and can be contained in another agent. The concept “Agent” of the current meta-model is not yet recursive. The third requirement states that an entity may need to be represented at the same time under multiple levels of representation. At present, an entity is represented in the model by one type of Agent, i.e., one level of representation.

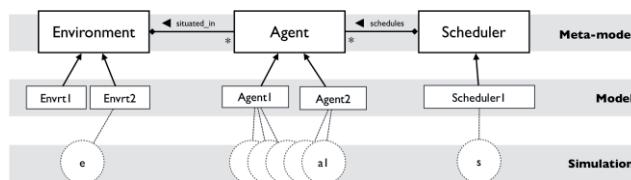


Figure 4. Meta-model of current agent-based modeling platforms (a simplified version). Squares represent classes whereas circles represent instances.

4. Generic Meta-model and “Morphogenesis” Operation

To support the development of multi-level agent-based model, we propose an extension of their meta-model (See Figure 4). We thus introduce three new *concepts* and a “morphogenesis” *operation*. These new concepts (*spatial scale*, *temporal scale*, *inner classes*²) represented as classes in the extended meta-model (See Figure 5) give an answer to the first two modeler’s requirements (R1 and R2). The “morphogenesis” operation, a particular change of class, (See Figure 6) fulfills the third modeler’s requirement (R3).

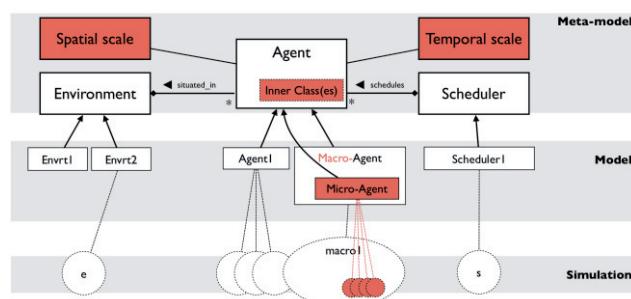


Figure 5. Extended meta-model supporting multi-level agent-based modeling (all new concepts are in red).

² In OO languages such as JAVA, an inner class describes a class declared entirely within the body of another class.

The “Spatial scale” and “Temporal scale” concepts, attached to Agent, offer the modeler the possibility to define the spatial scale and temporal scale for each type of agent. The “Spatial scale” concept defines an area on which agents can be situated. The “Temporal scale” concept defines how agents are scheduled in the simulation, e.g., their scheduling frequency and order. These two concepts make the meta-model capable of satisfying the first requirement R1. The “Inner class” concept is meant to meet the second requirement: R2. Thanks to this concept, the modeler can declare an Agent type inside another Agent type. An instance of the outer Agent type is called macro-agent while an instance of the inner Agent type is called micro-agent. A micro-agent always maintains a reference to its macro-agent. This concept thus facilitates the modeling of hierarchical organization of levels and the coupling between levels, i.e., the modeling of interactions and data exchanges between levels.

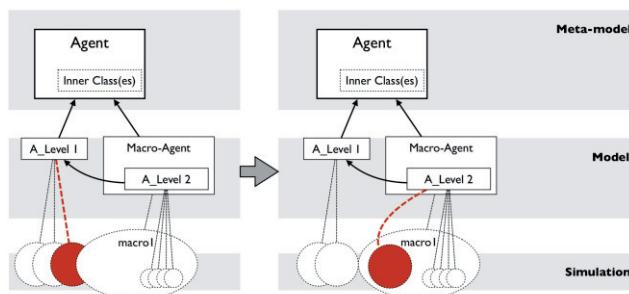


Figure 6. The “morphogenesis” operation corresponding to a change of class.
The instance of “A_Level 1” represented in red, changes its class to “A_Level 2”.

The “morphogenesis” operation enables an agent to change its level by changing its type. An entity can be represented in a model at different levels using different agent classes. The prerequisite condition of the operation is that there must exist a specialization link between two agent classes defining two representation levels of an agent. When the modeler applies this operation to change the representation level of an agent, under the hood, the following things happen: 1) a new instance of the agent type representing the new level of the agent is created; 2) shared attributes’ value between the agent types are copied from the (current) agent instance to the new agent instance; 3) the (current) agent instance is removed from the simulation and the new instance represents the new level of the agent in the simulation. As an agent type defines a level with associated spatial scale, temporal scale and behaviors, this operation thus allows the modeler to change the spatial scale, temporal scale and behaviors of agent dynamically during the course of the simulation.

Figure 6 illustrates this operation. The “Model” defines two possible representation levels of an agent using two agent classes: “A_Level 1” and “A_Level 2”. Adhere to the prerequisite condition, there is a specialization link between two levels: “A_Level 2” is a sub-class of “A_Level 1”. Before the operation, the agent, depicted as the red circle, is represented in the “A_Level 1” level, i.e., an instance of “A_Level 1” class. After the operation, the agent changes its level to “A_Level 2”, i.e., an instance of “A_Level 2” class. An important advantage of this approach is that introducing a new representation level, e.g., “A_Level 2”, requires no modification to the existing level, e.g., “A_Level 1”.

5. Implementation in a Java-based ABM Platform: GAMA

In order to operationalize the new concepts and the “morphogenesis” operation of the generic meta-model, we have decided to implement them in an agent-based modeling platform: GAMA. GAMA is a Java open source agent-based modeling platform which has been being developed by several Vietnamese and French research teams under the umbrella of UMI 209 UMMISCO/IRD since 2007. It has been being used to develop agent-based simulations, which support deciders in the management of environmental problem (flood control, mitigation of natural disasters, land-use and land planning, plant pests invasions, etc.). [12]

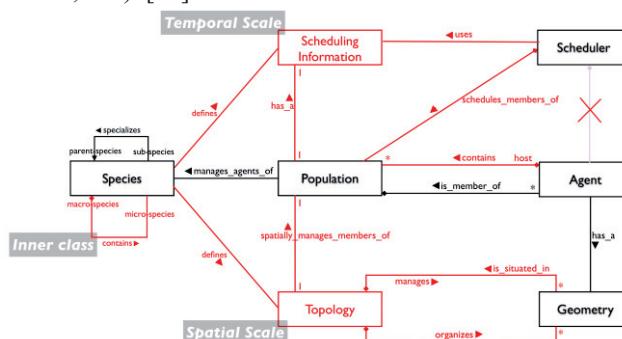


Figure 7. Multi-level meta-model of GAMA.

In order to make GAMA able to support the development of multi-level agent-based model, we enriched its meta-model, changed its simulation motor and enriched its modeling language (GAML). Due to the space limitation of the paper, we only present the modification of the meta-model (this section) and the enrichment of the GAML modeling language (next section).

Up to version 1.3, GAMA was underpinned by a meta-model, which was the same as the one of current agent-based modeling platforms. As presented in section 3, this meta-model did not support an easy development of multi-level agent-based model. We introduced the new concepts of the generic meta-model to the meta-model of GAMA. Figure 7 depicts the modification we made to the GAMA’s meta-model. Classes and lines in black color are existing parts, while classes and lines in red color are newly added parts to realize the concepts of the generic meta-model. We see that the “Spatial scale”, “Temporal scale” and “Inner class” concepts were appropriately implemented in the GAMA’s meta-model.

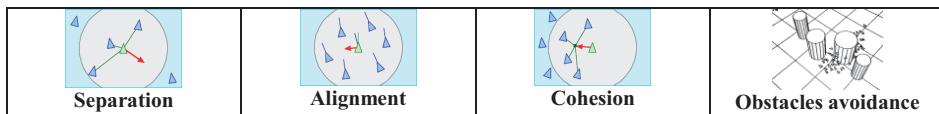
The “Species” concept defines attributes, behaviors of a class of same type agents. A “Population” is responsible for managing agents of a species. An “Agent” represents an agent of a species and belongs to a population. The “Topology” and “Scheduling Information” are the implementations of the “Spatial Scale” and “Temporal Scale” concepts. We implemented a set of predefined topologies, e.g., continuous, grid, graph, which are reusable by the modeler to define the spatial scale of agents. The “Scheduling Information” helps the modeler to specify how agents are scheduled in the simulation, e.g., scheduling frequency and order. To implement the “inner class” concept, we made a species able to contain other species and conversely to be contained by another species. The “morphogenesis” operation is implemented in reusable Java classes, which helps agent to change species (representation level) dynamically during the course of the simulation.

6. Developing a Multi-level Agent-based Model with GAML

GAML (GAMA Modeling Language) is the modeling language used to write agent-based model in GAMA. It provides the modeler with concepts highly adapted to agent-based modeling. Our enrichment enabled GAML to support the modeler in the development of multi-level agent-based model. A complete reference of this language can be found in the GAMA website [12]. This section presents how to develop a multi-level agent-based model with GAML.

We show below how to add up a new level of representation to an existing agent based-model: the Boids model. In 1986, Craig Reynolds made a model of coordinated animal motion such as bird flocks or fish schools basing on the following four steering rules: a) *Separation*: steer to avoid crowding local flockmates; b) *Alignment*: steer towards the average heading of local flockmates; c) *Cohesion*: steer to move towards the average position of local flockmates; d) *Obstacles avoidance*: steer to avoid obstacles in the environment. [13]

Table 1. The fours steering rules of boids model. Source: Reynolds [14]



Based on these four steering rules, we wrote the Boids model in GAML, which models the movement of birds in a continuous environment. We present here only some important code snippets of the model. To represent a class of same type agents, we use the “species” concept of GAML. We thus introduce the “boids” species within which we program four steering rules in the reflexes. A reflex is automatically executed by the simulation motor in each simulation step. Figure 8 shows the model’s conceptual structure (on the left) and the GAML code of “boids” species (on the right). “World” is a special species of a GAML model, which has a special built-in agent “world” playing the role of the global context of the simulation.

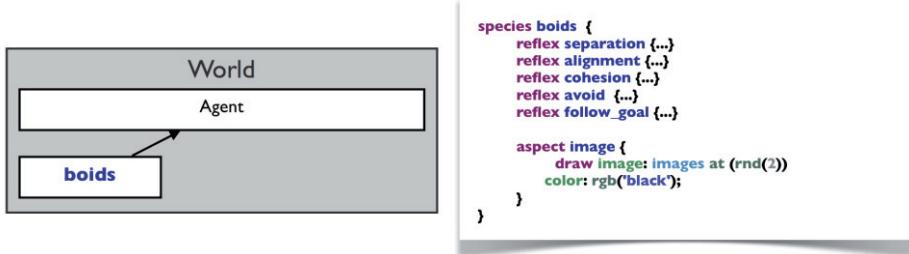


Figure 8. Original “Boids” model in GAML.

When running the simulation of this model, the modeler sees that nearby flying birds create flocks. In order to observe and understand the dynamics of the flocks, the modeler may desire to introduce a new kind of agent that represents these flocks. A flock will then be in charge of the movement of its underlying birds. “flock” agents will be created, destroyed, updated dynamically during the course of the simulation. To represent a flock of birds in GAML, the modeler introduces two species: “flock” and “boids_in_flock”. “boids_in_flock” is a micro-species of “flock” which represents the fact that a “flock” agent can contain “boids_in_flock” as micro-agents and facilitates the coupling between “flock” level and “boids_in_flock” level. Indeed, a “flock” agent

controls the movement of its underlying birds. Because “boids_in_flock” is a sub-species of “boids” it enables a bird to change its species from “boids” to “boids_in_flock”. This is done thanks to the morphogenesis operation and supports modeling the fact that a (free) bird changes behavior when it becomes the member of a flock.

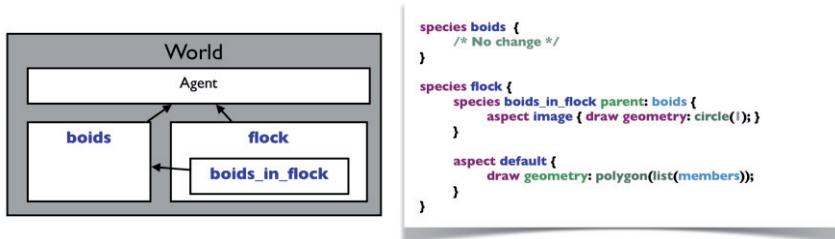


Figure 9. Multi-level “Boids” model in GAML. “boids_in_flock” is an micro-species of “flock” species.

Birds are now represented in the model at two levels of representation: “boids” and “boids_in_flock”. A bird can enter and leave flock dynamically during the course of the simulation. When a flock agent is created, particular nearby birds will become members of the flock and change their species from “boids” to “boids_in_flock”. A “flock” agent can thus capture nearby birds as micro-agents and change their species from “boids” to “boids_in_flock”. When a flock agent approaches an obstacle, it will disaggregate (die), its underlying birds will become free and change their species from “boids_in_flock” to “boids”. To model the fact that a bird can change its species from “boids” to “boids_in_flock” and vice-versa, the modeler uses the morphogenesis operation, which is implemented in GAML using the primitive statements “capture”, “release” and “migrate”. A comprehensive user guide of these statements can be found on the GAMA website [12]. The following two code snippets give examples of how the “capture” and “release” statements are used in this particular model. Code example 1 shows how a “flock” agent can capture some free birds as its member and change their species from “boids” to “boids_in_flock”. Code example 2 illustrates how a “flock” agent releases some member birds and changes their species from “boids_in_flock” back to “boids”. Due to space limitation of the paper, we do not give the snapshots of the simulation but this model is today included in the binary distribution of GAMA downloadable from [12].

Code example 1. GAML “capture” statement example

```
capture some_free_boids_agents as: boids_in_flock { ... }
```

Code example 2. GAML “release” statement example

```
release some_boids_in_flock_agents in: world as: boids { ... }
```

7. Conclusion and Perspectives

We have presented a generic approach for multi-level agent-based modeling. We first formalized a list of modeler’s requirements concerning multi-level representation in agent-based models using the BPH model. Then we have proposed a generic meta-model and a “morphogenesis” operation, which aim at satisfying these requirements. The generic meta-model extends the current agent-based meta-model with appropriate

concepts thus supporting multi-level representation in agent-based modeling. The “morphogenesis” operation enables agents to change their representation level dynamically during the course of the simulation. We then presented how the generic meta-model and the “morphogenesis” operation were together implemented in the GAMA platform. Using the Boids model, we have then showed that our proposal, implemented in GAMA, gives the modeler a straightforward approach to develop multi-level agent-based model.

As per today, our work has been being already used to support the development of two multi-level agent-based models. The first model is the BPH model presented in the beginning of this article. The second model deals with the organization of the crowd evacuation of the whole population of a coastal city in response to a tsunami first alert. The modeler did express the need to represent the dynamics of evacuees at two representation levels one being the agent-based and the second one an aggregated model represented by a mathematical approach [14].

To further confirm the genericity of our proposal, we intend to implement the generic meta-model and the “morphogenesis” operation on other agent-based platforms such as Repast and NetLogo [9, 10]. Further more, we would like to explore the problem of multi-host membership. This enables the modeler to represent that an agent can belong to several macro-agents at the same time. And we would like also to integrate our work with existing works on emergent detection techniques. As in a multi-level ABM, agents at higher levels can be *emergent structures* resulting from the interactions of other agents in the simulation that need to be automatically detected and reified during the course of simulation. In other words, our approach should also support modeling multi-level ABM starting with an initial level and incrementally adding new levels based on simulated virtual experiments. This is another direction of our current research.

References

- [1] E. Bonabeau, *Agent-based modeling: Methods and techniques for simulating human systems*, PNAS vol. 99 no. Suppl 3, May 14, 2002.
- [2] D.-A. Vo, *An operational architecture to handle multiple levels of representation in agent-based models*, Ph.D. thesis, University of Paris 6, 2012.
- [3] Vietnam Rice Export 2011. Yearly Export Statistics, Vietnam Food Association.
- [4] V.-A. Dyck, B. B. Misra, S. Alam, C. N. Chen, C. Y. Hsieh and H. S. Rejesus, *Ecology of Brown Plant Hopper in the tropics*. Los Banos (Philippines): International Rice Research Institute, 1997, pp. 61-98.
- [5] D. Servat, E. Perrier, J.-P. Treuil and A. Drogoul, *When Agents Emerge from Agents: Introducing Multi-Scale Viewpoint in Multi-agent Simulations*. MABS1998, LNAI 1534, pp. 183-198, 1998.
- [6] J.-G. Quijano, T. Louail and G. Hutzler, *From biological to urban cells: lessons from three multilevel agent-based models*. PRIMA 2010, LNAI 7057, pp. 620-635, 2012.
- [7] G. Morvan and D. Jolly, *Multi-level agent-based modeling with interaction reaction principle*. arVix: 1204.0634v1 [cs.MA] 3 April 2012.
- [8] S. Picault and P. Mathieu, *An Interaction-Oriented Model for Multi-Scale Simulation*. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.
- [9] M.J. North, T.R. Howe, N.T. Collier and J.R. Vos, *A Declarative Model Assembly Infrastructure for Verification and Validation*, in S. Takahashi, D.L. Sallach and J. Rouchier, eds. *Advancing Social Simulation: The First World Congress*, Springer Heidelberg, FRG (2007).
- [10] U. Wilensky, *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.
- [11] P. Taillandier, A. Drogoul, D.-A. Vo and E. Amouroux, *GAMA: a simulation platform that intergrates geographical information data, agent-based modeling and multi-scale control*. 13th PRIMA, India, 2012.

- [12] A. Drogoul and al, *The GAMA platform* – <http://gama-platform.googlecode.com>
- [13] C. Reynolds. Boids, Backgroud and Update - <http://www.red3d.com>
- [14] T.-N.-A. Nguyen, J.-D. Zucker, N.-H. Du and A. Drogoul, *A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks*. LNCS, Vol 7068, 2012, pp. 371-383.

Optimizing an Environmental Surveillance Network with Gaussian Process Entropy – An Optimization Approach by Agent-based Simulation

Viet Xuan TRUONG^{a,b,c,1}, Hiep Xuan HUYNH^b, Minh Ngoc LE^a,
 and Alexis DROGOUL^c

^a Faculty of Computer Science & Engineering, HCMUT, Ho Chi Minh City, Vietnam

^b DREAM Team/UMI 209 UMMISCO-IRD, Can Tho University, Vietnam

^c UMI 209 UMMISCO-IRD/UPMC, Bondy, France

Abstract. Finding an optimal design for the environmental surveillance network is a realistic need for any ecosystem manager. There are two main factors related to the optimization of a surveillance network: number of sampling points (deciding the sampling density) and locations of such sampling points. This paper aims at proposing an agent-based model to add k measuring devices into a current surveillance network. The simulation is used to verify multiple possibilities of a heterogeneous environment. A Correlation & Disk graph-based Surveillance Network (CDSN) is also implemented and then optimized. Gaussian process is used to model the measuring data and different covariance functions (e.g. variogram in geostatistics). The experimental results of the model are performed for the insect monitoring in Mekong Delta region, Vietnam.

Keywords. surveillance network, Gaussian process entropy, Agent-Based Model (ABM), geostatistics, Unit Disk Graph (UDG), Kriging standard error, optimal design, optimization

1. Introduction

Managing an ecosystem often requires a big budget for the surveillance network. Three main issues concerned by managers when building a surveillance network are: the number of measuring devices (depending on the establishing fund), the strategies of measuring (depending on the maintenance fund), and the data analysis techniques (depending on the research fund). Finding an optimal design normally considers all these related issues. The research aims to propose a new agent-based model [1][2] for optimizing an environmental surveillance network. In this paper, we concentrate on the optimal solution for adding k new measuring devices into a current surveillance network. Finding optimal locations is the main purpose of this article. By adding two different optimal conditions, the balance of the surveillance network and multiple local-constraint factors, this optimal model could be considered as an extension for a statistical model of optimal design (based on Gaussian process entropy).

To study the spatial structure by using multiple statistical methods (e.g. in geostatistics), the environment is often modeled as Gaussian processes (GPs). Krause

¹ Corresponding author: Viet Xuan TRUONG, Campus III, Can Tho University, 01 Ly Tu Trong street, Ninh Kieu district, Can Tho city, Vietnam; Email: txviet@ctu.edu.vn.

[3] listed three main ways for choosing sampling locations [4] (i.e. sensor locations), which are (1) geometry or disk model [5], (2) placing sensors at points of highest entropy in the GP model [3][6][7], and (3) optimal designs (e.g. A-, D- and E-optimal design) [8]. GP entropy is also applied in our approach as one of the most important conditions for the optimization strategy based on Kriging variance (Kriging variance is well-known as the GP entropy).

The light-trap network in Mekong delta region is used in the experiments where the Brown Plant Hopper [9] trapped-density is considered as monitoring variable. The current network gives a high variance in Kriging estimation [6][7]. Some research projects had been launched and some requirements for optimizing the existing network are already proposed, however the proposed solutions have not been accepted yet by the agricultural managers. These real experiments are applied in a small scale (space and time) and can't consider a range of related factors (e.g., rice transplantation regions, the weather, sea/river regions, etc).

This paper contains 7 main sections. Some important related works are introduced in Section 2. Section 3 presents the spatial structure and Gaussian processes. A short introduction of the Correlated & Disk graph-based Surveillance Network (CDSN) is presented in Section 4. In Section 5, an optimization process of surveillance network by using ABM is presented. The experiments and the simulation results are presented in Section 6. Section 7 summarizes the contribution and suggest some research directions.

2. Related works

Geostatistics is a branch of statistics that especially focuses on spatial analysis (instead of spatio-temporal analysis) [6][10][11]. Kriging (named of Krige, and developed by Matheron) is a popular interpolation to estimate the value of unknown locations from the set of sampling data, and applied in many research domains [11][12].

Kriging approach is also Gaussian process regression. Gaussian process is a collection of random variables, any finite number of which has a joint Gaussian distribution (Definition in [13] by Rasmussen). Krause et al. uses the map of variances as the information-driven key for adding new measuring devices, the location with highest variance will be selected for the new device, with the help of a mutual function as proposed in [3]. Bayesian interpolation [3][13] provides a standard set of procedures and formulas in GP regression (also GP classification).

In the design of experiments, optimal design is a kind of experimental designs that are effective with respect to some statistical criterion [3][8][14][15]. It allows estimating parameters without bias and minimum variance. Multiple optimal designs are proposed, such as the D-optimal design (minimizing the log-determinant), A-optimal design (minimizing the trace), and E-optimal design (minimizing the spectral radius of the error covariance), etc [8][15].

The UDG technique was first introduced by Clark [5] and has been used widely in ad-hoc communication [5]. In fact, there aren't many investigations of UDG in managing an ecosystem. An interesting research allows using the UDG as a bridge between an Individual-Based Model (IBM) and an Equation-based model (EBM), proposed in [16]. Some UDG-based research for estimating the BPH density and modeling the surveillance network were introduced in [17] and [18].

3. Spatial structure and Gaussian processes

In this research, we work on a surveillance network of n devices: $N = \{d_1, d_2, \dots, d_n\}$ at large time scale T . The measured data is defined as $Z = \{Z(s_1), Z(s_2), \dots, Z(s_n)\}$, where n is the current number of measuring devices and s_i denotes the site of device d_i . Our aim is to find how to add k new measuring devices into network N to get an optimal network $N_{OPT} = \{d_1, d_2, \dots, d_n, d_{n+1}, d_{n+2}, \dots, d_{n+k}\}$, N_{OPT} must give an optimal estimation (e.g. Kriging estimation) in comparison with N and other combinations of $(n+k)$ devices. This paper assumes that Z is measured from a random field $\{Z(s): s \in D \times T\}$, where D is related to R^2 of space and T is the time scale. Cressie [6] defined that this random field consists of two particles:

$$Z(s) = \mu(s) + \delta(s), s \in D \quad (1)$$

where $\mu(\cdot)$ denotes large-scale deterministic mean structure of the process and $\delta(s)$ denotes small-scale stochastic error structure that models spatial statistical dependence. There is an important assumption that $Z(\cdot)$ must be second-order stationary [6][11], this assumption is really necessary for multiple optimal linear estimators (e.g. Universal Kriging).

The question of our work is how to estimate $Z(\cdot)$ from their realization $Z = \{Z(s_1), Z(s_2), \dots, Z(s_n)\}$ with a minimal estimation variance.

The main purpose of any surveillance networks is to give the information about their observed variables, now modeled as a Gaussian process. In geostatistics, the variogram function is defined as the following:

$$2\gamma(x, y) = var(Z(x) - Z(y)) = E(|Z(x) - \mu(x) - (Z(y) - \mu(y))|^2) \quad (2)$$

The assumption of second-order stationarity is that $\mu(\cdot)$ must satisfy the constant-mean condition. Then, the variogram is defined in the equation below:

$$2\gamma(x, y) = E(|Z(x) - Z(y)|^2) \quad (3)$$

Gaussian process helps to evaluate the confidence of the Kriging estimation and also the Kriging variance. In probability theory and statistics, a Gaussian process is a stochastic process whose realizations consist of random values associated with every point in a range of time (or of space), so that each such random variable has a normal distribution [6] [13].

4. Correlated and Disk graph-based Surveillance Network (CDSN)

4.1. Motivation

Tobler defines the first law of geography as “*Everything is related to everything else, but near things are more related than distant things*” [19]. This law is usually mentioned in many geographical issues.

The network topology and intrinsic relations of the environment, if existing, are independently and inconsistently concerned by different researchers. Then the capacity of surveillance network is not dedicatedly considered and optimized. The Correlated & Disk graph-based Surveillance Network (CDSN) is proposed to overcome two big disadvantages of a normal surveillance process: (1) independence between the network topology and the intrinsic relations of environment, (2) discontinuity between the data collection process and the data analysis process. This approach is based on two important techniques used in sensor network (Unit Disk Graph) and data mining

(clustering by correlation), which help to establish the internal relations between different measuring devices.

A concrete case of CDSN was already developed in [18], in which we proposed an agent-based model named Unit disk graph-based Surveillance Network Model (USNM). This paper couldn't describe all characteristics of the CDSN, we just concentrate on some useful properties for our own optimization strategy: the balance of surveillance network and the estimation condition for adding new measuring devices.

4.2. Definition

A CDSN is represented as a directed and weighted graph $G = \langle V, E \rangle$ built from a set of vertices $V = \{v_1, v_2, \dots, v_n\}$, where each vertex is related to one measuring device in the network; and a set of edges $E = \{e_1, e_2, \dots, e_m\}$ determined by the relationship between two any different vertices via two connected functions: $D(v_i, v_j)$ and $C(v_i, v_j)$. $D(v_i, v_j)$ is defined as a spatial distance function and $C(v_i, v_j)$ is a correlation one. An edge $\{v_i, v_j\}$ belong to E if and only if $\{v_i, v_j\}$ satisfies both conditions of connection as the following:

- Condition of spatial distance: Spatial distance between two vertices v_i and v_j must less than or equal to a threshold of distance $\Theta_{spatialDistance}$. We have:

$$D(v_i, v_j) \leq \Theta_{spatialDistance} \quad (4)$$

where $\Theta_{spatialDistance}$ denotes the threshold of spatial distance (or disk radius) between two vertices.

- Condition of autocorrelation: Correlation coefficient of two vertices v_i and v_j must be greater or equal to a threshold of correlation $\Theta_{correlation}$.

$$C(v_i, v_j) \geq \Theta_{correlation} \quad (5)$$

where $\Theta_{correlation}$ denotes the threshold of sample correlation between two vertices.

The weighted and directed characteristics of CDSN aren't presented in this paper, the reader could find all related details in [18].

4.3. Some discussions about two conditions of connection

4.3.1. Condition of spatial distance

Clark [5] firstly proposed a spatial clustering technique named Unit Disk Graph technique to model the measuring network as a graph. All measuring devices having a distance under a specific threshold (denoting as the disk radius) will be connected together. And thus, only *near* measuring devices could be connected together, and it is used as the *first condition* of connection in CDSN.

This UDG-based surveillance network allows observing the spatial topology of all measuring devices. The balance of measuring devices is visualized and we could easily locate the area with high or low density of measuring devices.

4.3.2. Condition of autocorrelation

In CDSN, the correlation is used to determine the degree of similarity of collected data between two different measuring devices. This coefficient could diagnose the randomness of random field F [7][20]. If F is totally random, then the correlation between two different measuring devices is always equal 0. But following Tobler's law,

the exploratory data analysis shows that F isn't fully random, and some of them are well correlated.

Denoting the covariance function $C(\cdot)$ as the function of sample correlation between two sites s_1, s_2 (Pearson correlation is used in this research):

$$C(s_1, s_2) = \text{corr}_{\text{Pearson}}(Z(s_1), Z(s_2)) \quad (6)$$

In the simple case of univariate surveillance network, $Z(\cdot)$ becomes a scalar vector of time series data.

5. Optimization Model

In this section, CDSN is implemented as an ABM, where all elements are modeled as agents in this model: vertex agents, edge agents and network agent. We propose an agent-based optimization strategy to add k new devices into the current network of n devices. An optimization process with three optimal parameters is proposed, such as the Gaussian process entropy, the balance of network and local-constraint factors.

5.1. Three main parameters for optimization process:

5.1.1. Gaussian process entropy:

The map of Kriging standard errors is also considered as a map of Gaussian process entropies (the standard error is defined as the expectation of multiple Kriging standard deviations in a large temporal scale). Figure 1 shows the map of 50 light-traps of three provinces in Mekong delta region and the map of Universal Kriging standard errors (SE) of this network is showed in Figure 2.

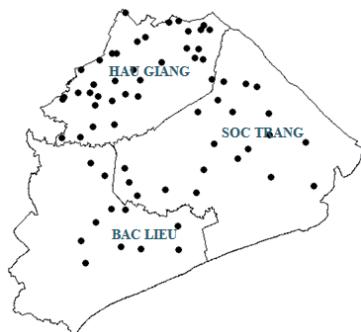


Figure 1. Measuring devices in three provinces: Hau Giang, Soc Trang and Bac Lieu, Mekong delta, Vietnam.

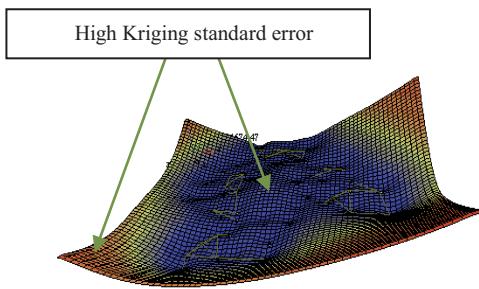


Figure 2. Map of Universal Kriging standard errors.

High standard errors are found in areas where a lack of measuring devices (center and bounds) is observed. Gaussian process entropy is defined by SE , and the priority is set higher if the entropy is greater.

5.1.2. Balance of network:

This parameter is constrained by the CDSN, it assumes that the new devices will be added around the most “isolated” vertices of the network (having the minimum degree). We only propose a solution to add a set of k new measuring devices $N_{new} = \{d_{n+1}, d_{n+2}, \dots\}$

$\dots, d_{n+k}\}$ into a current surveillance network with n devices $N_{current} = \{d_1, d_2, \dots, d_n\}$. In CDSN, each vertex agent is presented for a measuring device (e.g. the light-trap) and the edge is created if the following two conditions are satisfied: condition of spatial distance and condition of autocorrelation (Section 4).

The new vertex is located in one of neighbors of a selected vertex v_0 by a priority function. This priority takes into account the balance between the location into all other neighbor vertices (with a specific radius), and also the Kriging standard error at the local site.

$$\text{priority}(s) = \text{Balance}(s) * \text{KrigingSE}(s) \quad (7)$$

where

$$\text{Balance}(s) = \frac{\sum_{i=1}^{\text{number of neighbors}} \frac{d(s, N_i)}{\max(d(s, \forall \text{neighbors}))}}{\text{number of neighbors}} \quad (8)$$

KrigingSE(s) is the standard error at site s in the map of standard errors of Universal Kriging (Figure 2).

5.1.3. Local-constraint factors:

Table 1 shows a list of factors which are considered in our agent-based model. All these factors are combined into two important and standardized indices: *hinder index* and *attractive index*.

In the context of BPH monitoring, the attractive index reflects the convenient conditions for development of BPHs (natural or human affections). Otherwise, the hinder index could be used in two different roles. It's understood as an "opposite-index" of the attractive index, but also used to control the possibility of sampling placement (related to decision-supports of the managers). This approach uses the second role of the hinder index.

Table 1. Local-constraint factors

No	Name of factors	Hinder Index (HI)	Attractive Index (AI)
1	Regions of river & sea	1	0
2	Residential regions	1	0
3	Weather: humidity, temperature, etc.	(1 - AI)	[0..1]
4	Rice transplantation regions (by seasonal crops)	(1 - AI)	Winter-Spring: α ; Summer-Autumn: β ; All other: γ ($0 \leq \alpha + \beta + \gamma \leq 1$)

Determining the hinder index and attractive index from local constraint factors is almost heuristic. We can change them inside each simulation scenario. Historical data or expert experiences can be used to assign default values.

5.2. Optimization algorithm

Adding new vertices into an available CDSN by using optimization strategy is the main purpose of our optimization algorithm. This algorithm may be implemented in an agent-based model with some important agents:

- Cell agent: The grid contains the information about observed variables. There are $m \times n$ cell agents in the model, where n is the number of rows and m is the number of columns.
- UDG network (combinational agent): This agent contains two lists of sub-agents: vertices and edges. There is only one UDG network agent existing in the model.

- Local constraint agents: All factors in Table 1 are modeled and become various kinds of agent.

ALGORITHM: Addition of new vertices into a CDSN.

Inputs:

- *network*: Correlation & Disk Graph-based Surveillance Network with n vertices
- *k*: number of added vertices

Output:

- *network*: with $n + k$ vertices (CDSN)

Complexity: $O(n^2)$

BEGIN

```

1      i = 0;
2      LOOP (i <= k)
3          current_vertex = getMinimumDegree(network);
4          cells = getNeighborCells(current_vertex, radius);
5          potential_cell = NULL;
6          maximum_priority = 0;
7          FOREACH (current_cell in cells)
8              current_priority = getPriority(current_cell);
9              IF (maximum_priority < current_priority)
10                 potential_cell = current_cell;
11             END IF
12         END FOREACH
13         new_vertex = createNewVertexAt(potential_cell);
14         network.add(new_vertex);
15         i++;
16     END LOOP
17     RETURN network;

```

END

The **getMinimumDegree**(CDSN *network*) function returns a vertex which has the minimum degree (if there are more than one vertex found, then one of them could be returned in sequence). The **getPriority**(Cell *the_cell*) function is an implementation of Equation 7 in the model. The **getNeighborCells**(Vertex *vertex*, Integer *radius*) function returns all cells which have the distance to *vertex* less than or equal to *radius*. The **createNewVertexAt**(Cell *cell*) function is used to create a new vertex at location of *cell*.

The algorithm of optimization doesn't consider the third condition related to local constraints. To apply it, we can add the local condition into line 9:

IF (*maximum_priority* < *current_priority* **AND LOCAL_CONSTRAINTS**)

Local constraints can be heuristic and controlled by two combination indices: attractive and hinder (Table 1). For example, this condition can be defined as:

LOCAL_CONSTRAINTS = (*hinder_index* ≤ *threshold*)

5.3. Adding new vertex into the CDSN

To assure the consistency with the design of CDSN, the condition of autocorrelation [7][20] must be considered when adding new vertex s_* into the neighbor area of current vertex s_i . Krause [3] presented a correlation map of a specific location into all each others. We reuse this idea to determine $\hat{\rho}(s_i, s_*)$ in this research.

Denoting P_i as the vector of correlation coefficients from d_i to other ($n-1$) measuring devices.

$$P_i = \left\{ \rho_{i1}, \rho_{i2}, \dots, \rho_{i(i-1)}, 1, \rho_{i(i+1)}, \dots, \rho_{in} \right\} \quad (9)$$

Correlation map could be easily generated from this vector (by Kriging technique), where each vertex v_j keeps value of ρ_{ij} , and only vertex v_i keeps value of 1. Value of $\hat{\rho}(s_i, s_*)$ is the value of correlation map at location of s_* .

6. Simulation and experiments

Our simulation is developed in GAMA 1.5 platform [21] and some statistical operations are supported by the R language [22]. The Kriging estimation is applied with **gstat** library [23] and called in GAMA. We apply the experiments with data of 50 light-traps in 2010 in three provinces (Mekong delta, Vietnam). The BPH number is collected at night (by their attracted-by-light behavior). We call this number the *BPH trapped-density*. The time scale is set to one year, and a trapped-density vector of 365 elements is managed by each vertex agent in CDSN.

6.1. Description of simulation process

The simulation is used to verify multiple scenarios of local constraints for the optimization model, such as rice transplantation regions (in all crop seasons), the weather (which directly affects into the BPH behaviors), sea/river regions, etc. These constraints could be changed and considered in our optimization process.

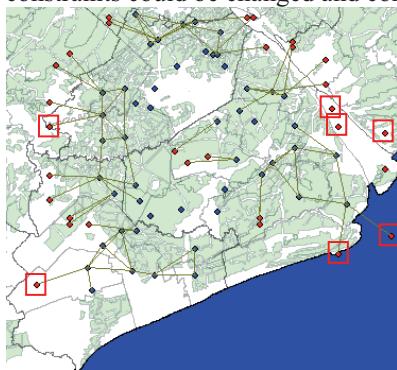


Figure 4. Optimization strategy with only two first parameters: Gaussian process entropy and Balance of network. The red squares mark the “infeasible” locations for light-traps.

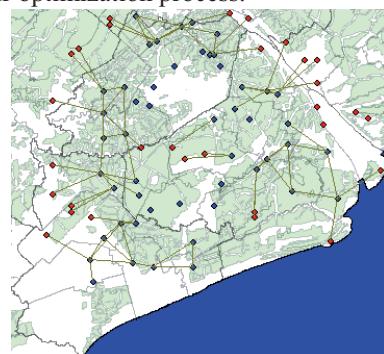


Figure 5. Optimization strategy with three parameters: Gaussian process entropy, Balance of network and Local-constraint factors.

Figure 4 shows the optimization strategy which only considers two first parameters of the optimization process: the *Gaussian process entropy* and the *balance of network*. The red circles are 30 new vertices (measuring devices) and the blue ones indicate 50 current vertices existing in the CDSN. All new vertices are located in the region of high standard error (center and bounds in Figure 2), but not necessary in the local maximum region.

In Figure 4, some new vertices are located in “infeasible” locations. Red squares mark seven new vertices (light-traps) which are located on the sea (2 vertices), on the river (2 vertices) and in regions without rice (3 vertices). These locations, of course, cannot be accepted by managers.

The same view of this study case is illustrated in Figure 5, where all “infeasible” vertex locations had been replaced by the others (by adding the local-constraints as Section 5.1.3). Managers could apply different constraints to get more experimental decision-supports. Local constraints are already added in the optimization.

7. Conclusion

This research is an extension of the statistical model of optimal design. A new agent-based model for optimizing an environmental surveillance network is developed with the help of Gaussian process. Gaussian process provides the best solution to estimate the posterior probability of a random field F , and then Universal Kriging variance is used as the deterministic parameter of the optimization strategy. With supports of Correlation & Disk graph-based Surveillance Network (CDSN), the balance of the network is considered as the second parameter. Also, multiple local-constraints are implemented to model the heterogeneity of the environment. These local-constraint factors are analyzed and standardized as two combinational parameters, hinder index and attractive index, helping to facilitate the optimization process.

We could take several local constraints into account in the optimization process (e.g. rice transplantation regions could be considered to add a new light-trap into an insect surveillance network). The important contribution is that multiple heterogeneous characteristics of the environment are easily embedded and considered in our research. Also, the simulation approach provides multiple “what-if” analyses for managers.

In conclusion, our contribution in this research is an ABM for optimization of a current surveillance network by adding k new measuring devices. Deleting or relocating the vertices isn’t concerned in this research. The assessment of performance for new optimal network is absent in this paper, but it’s already determined in a wide spectrum in the literature.

Acknowledgement

This publication was made possible through support provided by the IRD-DSF. The first author was funded by the PDI program (International Doctoral Program, website: <http://www.pdimsc.upmc.fr>).

References

- [1] J.-P. Treuil, A. Drogoul, J.-D. Zucker, P. Bourgine & É. Perrier: Modélisation et simulation à base d'agents : exemples commentés, outils informatiques et questions théoriques. Dunod, 2008. ISBN: 978-2-10-050216-5.
- [2] P. Auger, C. Lett & J.-C. Poggiale: Modélisation mathématique en écologie : cours et exercices corrigés. Dunod, 2010. EAN13: 9782100531929.
- [3] M. Köhl, S. Magnussen, M. Marchett: Sampling Methods, Remote Sensing and GIS Multiresource Forest Inventory. Springer, Heidelberg, (2006), 80–120. ISBN: 3-540-32571-9.
- [4] B. N. Clark, C. J. Colbourn, D. S. Johnson: Unit Disk Graph, *Discrete Mathematics* **86** (1990), 165–177.
- [5] N. A. C. Cressie, Statistics for Spatial Data. Wiley-Interscience, 1993. ISBN: 978-0-471-00255-0.
- [6] N. Cressie & C. K. Wikle: Statistics for Spatio-Temporal Data (Wiley Series in Probability and Statistics). Wiley, 2011. ISBN: 978-0-471-69274-4.
- [7] A. Krause, A. Singh, C. Guestrin: Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research* **9** (2008), 235–284.
- [8] A. Atkinson, A. Donev & R. Tobias: Optimum Experimental Designs, with SAS (Oxford Statistical Science Series). Oxford University Press, USA, 2007. ISBN 978-0-19-929660-6.
- [9] J.L.A. Catindig et al.: *Situation of planthoppers in Asia*. Planthoppers: new threats to the sustainability of intensive rice production systems in Asia. . Editors: K.L. Heong, B. Hardy. International Rice Research Institute (2009), 191–220.
- [10] G. Matheron, Principles of geostatistics, *Economic Geology* **58** (1963), 1246–1266.
- [11] H., Wackernagel, Multivariate geostatistics (Third edition). Springer-Verlag, Berlin (2003), 387p.
- [12] K.V. Huynh, J.-D. Zucker, X.H. Huynh, A. Drogoul: Spatial Estimator of Brown Plant Hopper Density from Light Traps Data. Proceedings on the IEEE-RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Ho Chi Minh City, Viet Nam. IEEE Catalog Number: CFP1256A-PRT (2011). ISBN: 978-4673-0308-8.
- [13] C. E. Rasmussen, C. K. I. Williams: Gaussian Processes for Machine Learning. MIT Press (2006), 243p. ISBN: 026218253X.
- [14] A. Warrick and D.E. Myers: Optimization of Sampling Locations for Variogram Calculations Water. *Resources Research* **23** (1987), 496–500.
- [15] D. J. Brus & G. B. M. Heuvelink: Optimization of sample patterns for universal kriging of environmental variables. Elsevier. *Geoderma* **138** (2007), 86–95.
- [16] N.D. Nguyen, et al.: Inferring equation-based models from individual-based models. In: PRIMA 2010 Conference (2010), 183–190.
- [17] X.V. Truong, X. H. Huynh, N.M. Le, A. Drogoul: Estimating the density of Brown Plant Hoppers from a light-traps network based on Unit Disk Graph, The 2011 International Conference on Active Media Technology (AMT 2011), Springer-Verlag, Berlin Heidelberg. *LNCS* **6890** (2011), 276–287.
- [18] X.V. Truong, X.H. Huynh, N.M. Le, A. Drogoul: Modeling a Surveillance Network based on Unit Disk Graph technique - Application for monitoring the invasion of insects in Mekong Delta Region, The 15th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2012), Kuching, Malaysia, Springer-Verlag, Berlin Heidelberg. *LNAI* **7455** (2012), 228–242 . DOI: 10.1007/978-3-642-32729-2_16.
- [19] W. Tobler, A computer movie simulating urban growth in the Detroit region. *Economic Geography* **46** (1970), 234–240.
- [20] P. Gouédard et al.: Cross-correlation of random fields: mathematical approach and applications. *Geophysical Prospecting* **56** (2008), 375–393. DOI:10.1111/j.1365-2478.2007.00684.x
- [21] Website: <http://code.google.com/p/gama-platform>
- [22] Website: <http://r-spatial.sourceforge.net/gallery/>
- [23] E.J. Pebesma, Multivariable geostatistics in S: the gstat package. *Computers & Geosciences* **30** (2004), 683–691. Website: <http://www.gstat.org/>

Coordination, Cooperation and Teamwork

Obligation and Prohibition Norms Mining Algorithm for Normative Multi-agent Systems

Moamin A. MAHMOUD^{a1}, Mohd Sharifuddin AHMAD^a, Azhana AHMAD^a,
Mohd Zaliman M. YUSOFF^a, Aida MUSTAPHA^b, Nurzeatul Hamimah Abdul
HAMID^c

^a*Universiti Tenaga Nasional, Kajang, Selangor, Malaysia*

^b*Universiti Putra Malaysia, Serdang, Selangor, Malaysia*

^c*Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia*

Abstract. Currently, research in normative multi-agent systems focus on how a visitor or new agent detects and updates its host norms autonomously without being explicitly given by the host system. In this paper, we present our proposed algorithm to detect the obligation and prohibition norms which we called the Obligation and Prohibition Norms Mining algorithm (OPNM). The algorithm exploits the resources of the host system, implements data formatting, filtering, and extracting the exceptional events, i.e. those that entail rewards and penalties of the obligation and prohibition norms and identifies the ensuing normative protocol. In this work, we assume that an agent is aware of its environment and is able to reason about its surrounding events. We then demonstrate the operation of the algorithm by applying it on a typical scenario and analyzing the results.

Keywords: Intelligent Software Agents, Norms Detection, Norms Mining, Normative Protocol Identification.

1. Introduction

Recently norms and normative systems have become the subject of interest of many researchers. As a term, norms are used to identify the behaviors of agents within a society and have been hailed as an efficient means to normalize agent behavior [1]. Basically, norms represent desirable behaviors for a community and generally understood as rules indicating what is expected to pursue i.e. obligation, prohibition, and permission, from a specific set of facts. However, to persuade agents to comply with norms, normative systems should be able to apply rewards or penalties [2].

The literature proposes three sorts of norms [3]. Regulative norms specify the ideal behavior of a system by means of obligations, prohibitions and permissions. Constitutive norms normalize the creation of institutional norms, in addition to the revision of the normative system itself [4]. Procedural norms are instrumental norms

¹ Corresponding Author: Moamin A. Mahmoud, Universiti Tenaga Nasional, Kajang, Selangor, Malaysia; E-mail: moamin84@gmail.com

[5], addressed to agents acting roles in the normative system, intending to perform the social order particularly in terms of substantive norms [6].

An agent is a normative agent when its behavior is regulated by the obligations it must comply with, prohibitions that limit the type of goals it can pursue and social commitments that have been created [7]. While a multitude of research work address norms regulation in normative frameworks [e.g. 7, 8, 9], very few have been found to discuss how the agent can comply with the normative system protocol without the need of doing that offline [10]. The related works in this area report a few results [10, 11, 12, 13].

In this paper, we present our proposed algorithm to detect the obligation and prohibition norms that entail reward or penalty, which we called Obligation and Prohibition Norms Mining algorithm (OPNM). The algorithm consists of several steps, which are data formatting that groups data into two classes of events, reward and penalty events; filtering the grouped data and identifying the final set of data for each group; and extracting the obligation and prohibition norms from the final sets. When a visitor agent detects the local agents getting rewards or penalties for some actions (i.e. exceptional events), it exploits the resources of the host system and uses the OPNM algorithm to implement data formatting, filtering, and extracting the exceptional events (of the obligation and prohibition norms) and identifying the normative protocol. In this work we assume an agent is aware of its environment and is able to reason about the surrounding events.

The term “Norms Mining” was coined by Mahmoud et.al. [14], from the concept of data mining. In their work, they used the term to mine the norms from a domain’s log file and extract the potential norms of the domain. In this paper, we use this term to mine the obligation and prohibition norms.

This paper is organized as follows: Section 2 dwells on the related work on norms identification and emergence. In Section 3, we present the conceptual framework for our research on norms detection. Section 4 discusses an example to explain the norms detection approach using the norms mining technique and Section 5 analyzes the results of the norms mining example. Section 6 concludes the paper.

2. Related Work

Norms and normative systems have received much attention due to their ability to coordinate agents’ interactions [15, 7]. Generally, norms are integrated in an agent system to guide it to behave in a socially harmonious way in a multi-agent society.

From the architectural standpoint of normative multi-agent systems [16], the literature provides ample research information on normative systems. Broersen et al. [17] propose the belief, obligation, intentions and desire (BOID) architecture that has an obligation component, O, added to the original BDI architecture. Ahmad et al. [16] developed a normative agent framework called the Obligation-Prohibition-Recommended-Neutrality-Disliked (OP-RND) Framework to regulate rules and norms effectively. Their agents perform the tasks from a set of pre-compiled tasks based on their beliefs of the reward and penalty associated with the selected tasks [16].

In the detection of normative multi-agent systems, the related researches in this area reveal a few works [10]. Campos et al. [18] uses a norm adaptation mechanism based on social power, which consists of providing support for the coordination of agents. A

generic level assisted architecture is used to develop systems that self-adapt their organization depending on their evolution. Mahmoud et al. [19] developed a norm mining technique for a visitor agent to detect the norms of a community of local agents to comply with the community's normative protocol. The visitor agent is equipped with an algorithm, which detects the potential norms from a domain's log file.

In another related work, Symeonidis et al. [20] developed K-Profile, an agent-oriented algorithm that deals with agent actions. The algorithm predicts agent behaviors and actions that are adapted from data mining technique. It exploits data mining techniques to extract knowledge from historical data, which express the actions of agents within the multi-agent system. They use this mechanism to find out action profiles and offer recommendations on agent actions which could be a good factor to enhance agent behavior [20].

In a similar work, Savarimuthu et al. [9, 11] propose a norm identification technique in which an agent infers the norms of a community without the norms being explicitly given to the agent. Their technique uses association rule mining to identify the tipping norm in an agent-based simulation of a virtual restaurant. In this scenario, they assume that agents are located in the restaurant where other agents entering the restaurant may not be aware of the protocol associated with ordering and paying for food items and other associated norms.

3. The Obligation Prohibition Norms Mining (OPNM) Algorithm

For an agent to detect the obligation and prohibition norms, it needs to collect data for analysis. In social learning theory, according to [10] based on [21, 22, 23], an agent can learn new behavior through observation of punishment or rewards and by monitoring of agent actions. They noted that social learning theory can be used to detect the norms within an agent society. Consequently, in this algorithm, we exploit the exceptional events, i.e. those that entail rewards and penalties.

We use the approach of signaling events [10] that assumes that an agent is able to sense its surrounding events and recognize rewarded and penalized actions. A rewarded action represents a positive signal and penalized action represents a negative signal. A positive signal can be any type of rewards e.g. thanking with a smile, while a negative signal can be any type of punishment e.g. shouting.

The agent starts collecting data when it observes a positive or negative signal by gathering the related events and adding to its record file. The agent stops collecting when repeated events are encountered. Figure 1 shows the architecture of OPNM algorithm.

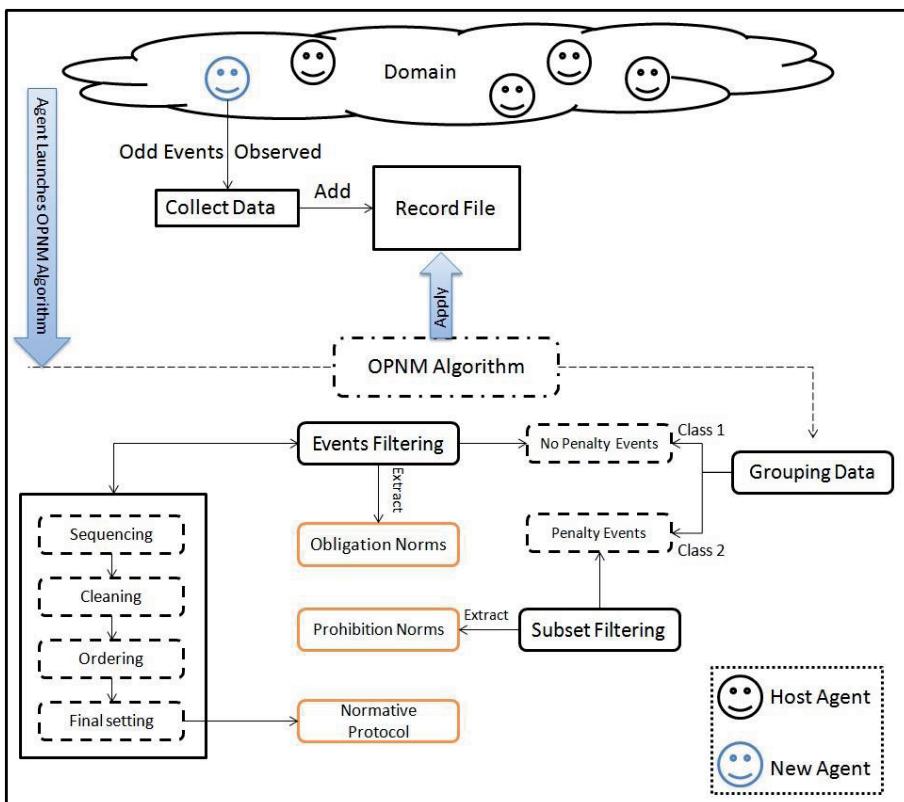


Figure 1. The OPNM algorithm architecture

When an agent observes a reward or penalty action, it launches the OPNM algorithm after it completes the data collection phase. The algorithm consists of several steps, which are:

1. Grouping data into two classes:
 - (a) Class 1 contains no penalty events.
 - (b) Class 2 contains penalty events.
 2. Events filtering is applied on Class 1 (i.e. no penalty events) by applying:
 - (a) Sequencing - labeling each event sequence from 1 to n (1, 2, 3, .., n). For example, if the sequence of actions of an agent, Agent1: (a, b, c, d), then applying labeling it becomes Agent1: (a_[1], b_[2], c_[3], d_[4]).
 - (b) Cleaning - selecting the highest number of sequence between the repeated actions in Class 2 and removing the rest. For example, suppose there are two events:
 $a_{[1]}, b_{[2]}, c_{[3]}, d_{[4]}$
 $b_{[1]}, d_{[2]}$
- From these two events, remove b(1) and retain b(2) and so on.
- (c) Ordering - after getting a set of the highest number of sequence for non-repeated actions, order the set from 1 to n (count up).
 - (d) Final setting - get the final set of Class 1 (F-Class1) after ordering to use it for the next step.

3. Subset filtering is applied on Class 2 by removing the subset events which has same sequence of events in a bigger set. For example, from the Venn diagram shown in Figure 2, we recognize two sets, P and Q: the first set is $P=\{b, c\}$ and the second set is $Q=\{a, b, c, d\}$. We also recognize that the first set is a subset of second set and because if the set is true then the subset is true. “It is always the case that two compound statements are equivalent if and only if they have the same truth sets” [24]. Therefore, we remove all the actions, which are subset of the bigger set or subset of itself if it has the same sequence of a bigger set.

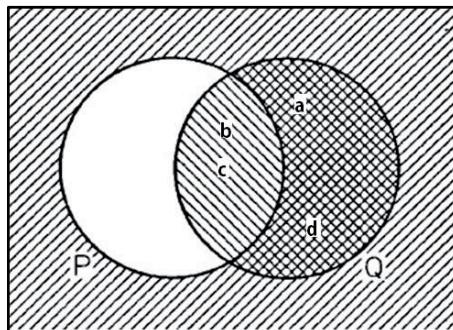


Figure 2. Venn diagram

Mathematically, we can represent subset filtering as follows:

$$\text{Filter}(\text{agentX}, \text{setB}) \Leftrightarrow (\text{A} \cap \text{B} = \text{B} \rightarrow \text{B} \subseteq \text{A})$$

$$\text{Filter}(\text{agentX}, \text{setA}) \Leftrightarrow (\text{A} \cap \text{B} = \text{A} \rightarrow \text{A} \subseteq \text{B})$$

After completing the subset filtering, the agent removes all the subsets from Class 2 and gets the final set of Class 2 (F-Class2).

4. Extracting the norms that caused the exceptional events. The obligation norms must appear in F-Class1 because it is the storage of non-penalty events, and the prohibition norms must appear in the F-Class2 which contains the penalty events.

- Obligation Norm: It is the norm that if it is not exercised by an agent, the agent is penalized. The agent can detect an obligation norm by finding the set-theoretic complement between the actions of F-Class2 and F-Class1 that have the same first until the last action. If we refer to F-Class1 as C_1 , a segment of event (or actions) of Class1 as C_{s1} , Class 2 as C_2 and Obligation Norm as O, then,

Extracting the Obligation Norm,

$$\text{If } ((C_{s1} \setminus C_2) = \emptyset) \Rightarrow (O = \emptyset) \dots \dots \dots \quad 1$$

$$\text{else } ((C_{s1} \setminus C_2) = N) \Rightarrow (O = N) \dots \dots \dots \quad 2$$

Which means that if set C_{s1} without set C_2 equal \emptyset , then no obligation norm appear in C_{s1} , otherwise, the results represent the Obligation norm.

- Prohibition norm: It is the norm that if it is exercised by an agent, the agent is penalized, but not otherwise. The agent can detect a prohibition norm by

finding the set-theoretic complement between the actions of F-Class2 and F-Class1. If we refer to the Prohibition Norm of penalized events as P, then Extracting the Prohibition Norm,

$$\text{If } ((C_2 \setminus C_{S1}) = \emptyset) \Rightarrow (P = \emptyset) \dots \dots \dots \quad 3$$

$$\text{else } ((C_2 \setminus C_{S1}) = N) \Rightarrow (P = N) \dots \dots \dots \quad 4$$

which means that if set C_2 without set C_{S1} equal \emptyset then no prohibition norm appear in C_2 , otherwise, the results represent the prohibition norm of penalized events.

5. Identifying the normative protocol: An agent can identify the normative protocol from F-Class1 because F-Class1 represents the set of norms that are not penalized.

4. An OPNM Algorithm Example

To illustrate the algorithm and to test its validity, we present an example of agent actions in an elevator. The aim is to identify the missing norms and recognize between obligation and prohibition norms.

In this scenario, we define a set of norms which consists of the actions, arrive (a), wait (w), enter (e), push (p), smoke (s), excuse (x), leave (l). We assume that some agents, (A, B, C,...), are penalized, π , by a punisher agent, α , for failing to wait, pushing or smoking.

We assume the following events are in the record file, which have been collected by the agent, represented only as symbols:

- (A) = (e, x, l)
- (B, α) = (w, e, s, π)
- (C) = (a, w, e, x, l)
- (D, α) = (a, e, l, π)
- (E) = (w, e, x)
- (F, α) = (e, p, l, π)
- (G) = (a, w, e)
- (H) = (x, l)
- (I, α) = (a, w, e, s, e, l, π)

Step 1: Grouping the data into two classes.

- Events with no penalties are grouped in Class1
 - (e, x, l)
 - (a, w, e, x, l)
 - (w, e, x)
 - (a, w, e)
 - (x, l)
- Events with penalties are grouped in Class2
 - (e, p, l, π)
 - (a, e, l, π)
 - (a, w, e, s, x, l, π)

(w, e, s, π)

Step 2: Filtering of Class 1 events

- Sequencing is implemented by labelling each action in a sequence. All Class1 events are labelled as follows:
 - (e_[1], x_[2], l_[3])
 - (a_[1], w_[2], e_[3], x_[4], l_[5])
 - (w_[1], e_[2], x_[3])
 - (a_[1], w_[2], e_[3])
 - (x_[1], l_[2])
- Cleaning is performed by selecting the highest number of sequence between the repeated actions. Form From the above set of events, the following event is retained.
 - (e_[3], x_[4], l_[5], a_[1], w_[2])
- Ordering entails rearranging the actions in ascending numerical order. The following order is produced.
 - (a_[1], w_[2], e_[3], x_[4], l_[5])
- Final set (F-class1) is obtained.
 - (a, w, e, x, l)

Step 3: Subset filtering of Class 2 by removing the subset events which has same sequence of events for bigger set.

(e, p, l)
 (a, e, l)
 (a, w, e, s, x, l)
 (w, e, s)

Filtering F-Class2, we obtained the following event sets in which the last event (w, e, s) is a subset of the third (a, w, e, s, x, l) event. Let

$$\begin{aligned} E_1 &= (e, p, l) \\ E_2 &= (a, e, l) \\ E_3 &= (a, w, e, s, x, l) \end{aligned}$$

Step 4: Extracting the norms that caused the exceptional events. If E = Event, C₁ = F-Class1, C_{S1} = a segment of event Class1, C₂ = F-Class 2, N_O = Obligation Norm, and N_P = Prohibition Norm, then,

- Obligation norm, N_O, is $((C_{S1} \setminus C_2) = N) \Rightarrow (N = N_O)$ and
 - Event1 C₂
 $E_1 C_2 = (e, p, l)$
 $C_{S1} = (e, x, l)$
 Therefore,
 $((C_{S1} \setminus E_1 C_2) = x) \Rightarrow (x = N_O)$
 - Event2 C₂
 $E_2 C_2 = (a, e, l)$
 $C_{S1} = (a, w, e, x, l)$
 Therefore,

$$((C_{s1} \setminus E_2 C_2) = (w, x)) \Rightarrow ((w, x) = N_o)$$

- Event3 C_2
 $E_3 C_2 = (a, w, e, s, x, l)$
 $C_{s1} = (a, w, e, x, l)$
Therefore,
 $((C_{s1} \setminus E_3 C_2) = \emptyset) \quad (N_o = \emptyset)$

- Prohibition norm, N_p , is $((C_2 \setminus C_{s1}) = N) \Rightarrow (N = N_p)$ and
 - Event1 C_2
 $E_1 C_2 = (e, p, l)$
 $C_{s1} = (e, x, l)$
Therefore,
 $((E_1 C_2 \setminus C_{s1}) = p) \Rightarrow (N_p = p)$
 - Event2 C_2
 $E_2 C_2 = (a, e, l)$
 $C_{s1} = (a, w, e, x, l)$
Therefore,
 $((E_2 C_2 \setminus C_{s1}) = \emptyset) \Rightarrow (N_p = \emptyset)$
 - Event3 C_2
 $E_3 C_2 = (a, w, e, s, x, l)$
 $C_{s1} = (a, w, e, x, l)$
Therefore,
 $((E_3 C_2 \setminus C_{s1}) = s) \Rightarrow (N_p = s)$

The OPNM algorithm detected wait (w) and excuse (x) as obligation norms and push (p) and smoke (s) as prohibition norms.

Step 5: Identifying the normative protocol. An agent can identify the normative protocol from F-Class1 because this class does not contain penalized actions. From Class 1, the normative protocol is (arrive, wait, enter, excuse, leave).

5. Analysis of Results

In this section, we discuss the result of elevator scenario. Although the given example is not enough to measure the capability of algorithm accurately, it illustrates the challenges of regulative norm mining technique and the potential solution that could be applied via the OPNM algorithm.

From elevator scenario, we observe the following points:

1. The example shows clearly that the algorithm succeeded in identifying the two types of norms: obligation and prohibition.
2. The algorithm succeeded in identifying multiple norms. The example shows that the algorithm detected four norms for the two types: two norms detected are of obligation types and two norms are prohibition norms.

3. The algorithm has also successfully identified the normative protocol in of the example.

6. Conclusion and Further Work

The progress of several software agent capabilities, such as agent collaboration, negotiation, trust, and mobility needs additional approach to agents' verification and authentication. At the society level, norms could offer another line of defence by constraining agents to comply with the society's norms.

In this research, we present obligation and prohibition norms mining algorithm as a new technique to extract a domain's obligation and prohibition norms. The algorithm could provide a suitable solution for software agent and robot communities to comply with and adapt to the new environments and learn new behaviours which lead to improved capabilities.

From the given example, we demonstrated the OPNM algorithm that entails the process of data formatting, filtering and extracting the different types of norms and normative protocols. The results show that the OPNM algorithm successfully detected the obligation and prohibition norms and identify the normative protocol.

In our further work, we shall build a virtual environment that has several domains each with a finite number of agents and make further tests on the OPNM algorithm to discover other aspects such as the time taken for the agent to detect the norms and the influence of the number of agents on the speed of detection.

Acknowledgement

This project is sponsored by the Malaysian Ministry of Higher Education (MOHE) under the Exploratory Research Grant Scheme (ERGS) No. ERGS/f/2011/STG /UNITEN/02/8.

References

- [1] M. Alberti, A. S. Gomes, R. Goncalves, J. Leite, and M. Slota, Normative systems represented as hybrid knowledge bases, *Proceedings of the 12th International Conference on Computational Logic in Multi-agent Systems, CLIMA'11*, 2011.
- [2] R. Gonçalves, J. J. Alferes, Specifying and reasoning about normative systems in deontic logic programming, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems , AAMAS'12*, 2012.
- [3] P. Caire, A normative multi-agent systems approach to the use of conviviality for digital cities. *Proceedings of the international conference on Coordination, organizations, institutions, and norms in agent systems III, COIN'07*, 2007.
- [4] G. Boella, L.V.D. Torre, Regulative and constitutive norms in normative multiagent systems, *In Procs. of 9th International Conference on the Principles of Knowledge Representation and Reasoning*. Whistler (CA), 2004.
- [5] G. Boella and L.V.D. Torre, Substantive and procedural norms in normative multiagent systems, *In Proceedings of J. Applied Logic*, 2008, 152-171.
- [6] Merriam-Webster, Dictionary of Law, Merriam-Webster, 1996.
- [7] C. Hollander, A. Wu, The current state of normative agent-based systems, *Journal of Artificial Societies and Social Simulation* 14 (2) 6, 2011.

- [8] C. Castelfranchi, F. Dignum, C. Jonker, J. Treur, Deliberative normative agents: principles and architecture. *The 6th International Workshop on Intelligent Agents, Agent Theories, Architecture and Languages (ATAL09)*, Orlando, Florida, 1999.
- [9] B. T. R. Savarimuthu, M. Purvis, S. Cranefield, M. Purvis, How do norms emerge in multi-agent societies? mechanisms design. *The Information Science Discussion Paper Series–2007, University of Otago*, 2007.
- [10] B. T. R. Savarimuthu, S. Cranefield, M. Purvis, M. Purvis, Obligation norm identification in agent societies. *Journal of Artificial Societies and Social Simulation*, 2010.
- [11] B. T. R. Savarimuthu, S. Cranefield, M. Purvis, M. Purvis, Internal agent architecture for norm identification, *The International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems*, pp. 156-172, 2009.
- [12] G. Andrijghetto, R. Conte, P. Turrini, M. Paolucci, Emergence in the loop: simulating the two way dynamics of norm innovation. in: *Normative Multi-agent Systems*, in *Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI)*, Schloss Dagstuhl, Germany, 2007.
- [13] J. Campos, M. López-Sánchez, M. Esteva, A case-based reasoning approach for norm adaptation, *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS'10)*, Springer, Spain, 2010.
- [14] M. Mahmoud, , M. S. Ahmad, A. Ahmad, , M. Z. M. Yusoff, , A. Mustapha, Norms detection and assimilation in multi-agent systems: a conceptual approach, *D. Lukose, A.R. Ahmad, and A. Suliman (Eds.): KTW 2011, Communications in Computer and Information Science*, Volume 295, Springer-Verlag Berlin Heidelberg 2012.
- [15] Y. Shoham, M. Tennenholtz, On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.
- [16] A . Ahmad, An agent-based framework incorporating rules, norms and emotions (Op-Rnd-E), *PhD thesis, College of Graduate Studies, Universiti Tenaga Nasional*, January 2012.
- [17] A. Ahmad, M. Ahmed, , M. Z. M. Yusoff, M. S. Ahmad, , A. Mustapha, Resolving conflicts between personal and normative goals in normative agent systems, *The Seventh International Conference on IT in Asia 2011 (CITA 2011)*, Kuching, Sarawak, July 2011.
- [18] J. Broersen, M. Dastani and L. v. d. Torre (2001). Resolving conflicts between beliefs, obligations, intentions, and desires: symbolic and quantitative approaches to reasoning with uncertainty, *6th European Conference, ECSQARU, Toulouse*, France, Springer.
- [19] J. Campos, M. López-Sánchez, M. Esteva, A case-based reasoning approach for norm adaptation, *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS'10)*, Springer, Spain, 2010.
- [20] M. Mahmoud, M. S. Ahmad, A. Ahmad, M. Z. Mohd Yusoff, A. Mustapha, A norms mining approach to norms detection in multi-agent systems, *International Conference on Computer & Information Sciences*, Kuala Lumpur, Malaysia, 2012.
- [21] A. Symeonidis and P. Mitkas, A methodology for predicting agent behavior by the use of data mining techniques. In *Autonomous Intelligent Systems: Agents and Data Mining*, pp. 161–174, Lecture Notes in Computer Science 3505, Springer Berlin / Heidelberg, St. Petersburg, Russia, June 6-8 2005.
- [22] A. Bandura, Social learning theory, *Prentice Hall*, 1977.
- [23] R. Conte, , F. Dignum, From social monitoring to normative influence. *Journal of Artificial Societies and Social Simulation*, JASSS, Vol. 4, No. 2, (2001).
- [24] J. G. Kemeny, J. L. Snell, G. L. Thompson, and P.Doyle, Finite mathematics, mathematics at dartmouth, 5 October 1998. <http://www.math.dartmouth.edu/~doyle/docs/finite/finite.pdf>.

Role and Member Selection in Team Formation Using Resource Estimation

Masashi HAYANO ^{a,1}, Dai HAMANO ^a and Toshiharu SUGAWARA ^a

^aDepartment of Computer Science and Engineering,
Waseda University, Tokyo 1698555, Japan

Abstract. We propose an efficient team formation method for multi-agent systems consisting of self-interested agents in task-oriented domains. Services computing on computer networks have been rapidly increasing. Efficient team formation for service tasks is considered to be a way to improve performance. Our method is based on our previous parameter learning method enabling agents to efficiently form teams but requiring prior knowledge about all others' resources. We extended that method by adding a resource estimation method so as to increase its applicability to actual application systems. We experimentally evaluated our method by comparing it with the previous method and the task allocation using contract net protocol (CNP). The results demonstrated that the proposed method outperformed other methods even though it did not require prior knowledge about resources in other agents. We discuss the reason for this improvement.

Keywords. Team Formation, Task Allocation, Learning

1. Introduction

The task and resource assignment problem in multi-agent systems (MAS) is an abstract problem framework for many applications in networked distributed systems. For example, services on the Internet are often constructed by combining multiple service elements in a bottom-up manner, and these elements are allocated to and processed in a number of nodes that have suitable capabilities for the required service elements [1]. The execution of these services can be seen as cooperative collective activities by multiple agents, in which an information service can be abstracted as a task consisting of a number of subtasks that corresponds to a service element [2]. These subtasks should then be allocated to agents that have sufficient resources and capabilities. The execution of tasks by allocating their subtasks to appropriate agents is often referred as the *team formation* problem in a multi-agent systems context. A service cannot be provided even if only one of its service elements is processed with a significant delay. This means efficient and effective, as well as safe, assignment/allocation in a distributed environment is key to providing timely and steady services.

However, there are a number of difficulties with team formation in these kinds of applications. First, agents have their own resources and functions that have been designed

¹Corresponding Author: Masashi Hayano, Department of Computer Science and Engineering, Waseda University, Tokyo 1698555, Japan; E-mail: m.hayano@isl.cs.waseda.ac.jp

by different developers. Second, how many and what tasks will be requested vary over time because service requests are reflections of human activities. Furthermore, the machines that the agents run on may asynchronously be replaced by higher-performance machines or the agent programs may be upgraded with new capabilities. Thus, it is impossible to anticipate what teams of agents will be appropriate in the design phase of the system. Nevertheless, the systems have to process a huge number of tasks. This means one must have efficient and adaptive methods of team formation in dynamic environments in order to use the full capabilities of all agents in the system.

This paper addresses the issue of forming teams efficiently and effectively by self-interested agents in a task-oriented domain, where teams receive the utilities associated with tasks after they have been formed successfully. For this purpose, we have already proposed a method of implementing the aforementioned functionality for evolving adaptive behaviors [3]. In the model of team formation presented in [3], an agent autonomously selects the role of a *leader* or a *member* that it would play in a team. To decide the role and which member agents should be invited as its members, we use three learning parameters that express the utilities that an agent could expect to set as a member, the utilities it could obtain as a leader, and the expected values for which an agent will accept the solicitation to join the team. Thus, agents can decide their activities from the two-sided viewpoint of being a leader and a member by learning these parameter values. Although such learning can improve the overall efficiency of the team formation process, the method assumes that each agent knows the resources given to all agents. This assumption is too restrictive to apply to actual applications.

To overcome this restriction, we extended our learning method so that all agents can estimate the resources of other agents through the team formation process. We experimentally showed that this method can receive more utility than the previously proposed one [3]. We also compared our method with the contract net protocol (CNP) [4,5], which is a well-known task allocation protocol based on the market model and does not assume that agents know the resources and capabilities of other agents.

This paper is organized as follows. We discuss related work in Section 2 and the models for team formation in Section 3. In Section 4, we present the parameter learning method proposed in [3] and explain how our resource estimation method is incorporated in it. Section 5 explains how we experimentally evaluated our method. Our method could be slightly better than our previous method and the CNP, although the proposed method does not require resource information of other agents. Finally, we discuss the reason for the improvement in our method although it is essentially based on the previous method.

2. Related Work

Team formation is a key issue in the literature on MAS, and there have been a number of studies on coalition formation and team formation from the viewpoint of the task/resource allocation problem [6,7]. For example, Shehory and Kraus [6] proposed a polynomial-time algorithm for coalition formation to calculate Pareto-optimal solutions. Yang and Luo [7] discovered an optimal team corresponding to agents and multiple tasks using a genetic algorithm. However, their research assumed that the agents' internal states and the tasks were known. It is almost impossible to anticipate the resources needed for tasks and the resources/functions of each agent in an open system.

Abdallah and Lesser proposed an effective (polynomial-time) method of team formation using reinforcement learning in tree-structured agent networks [8]. Teams are efficiently formed because all the agents in the network learn the allocation policies for tasks sent to other agents at lower levels of the hierarchy from past experiences. Katayanagi and Sugawara extended this method by reorganizing agent networks so that they would allocate more tasks to agents that become idle after learning [9]. This extended learning method improved the efficiency of the team formation process. However, its applications are limited since its agent network needs to be hierarchical.

The method of learning for efficient team formation in a task-oriented domain developed by Genin and Aknine [10] is related to our proposal. In this method, all agents record which agents proposed each task T and which agents accepted or rejected T . The agent acting as a leader selects team members by referring to this history. However, the agents only learn which others are likely to accept the team members for the task set containing T ; they do not learn which roles (a leader or a member) they should play or which roles could increase their earned utilities. We proposed a method of learning in which agents learned which role, leader or member, would be better for earning more utilities [3]. They found that it outperformed the method proposed in Ref. [10]. However, these studies assumed that all resources in agents are known to each other *a priori*. The aim of this study is to devise a learning method that does not require prior knowledge of the agents' resources, by incorporating a resource estimation method into the learning of the team formation process proposed in Ref. [3].

3. Model and Problem Description

Here, we describe the model consisting of agents, tasks, and team formation. Note that this is almost identical to those described in Ref. [3]. However, we assume that a task consists of a number of subtasks, whereas a task is singleton in Ref. [3]. We believe that this extension is required in order to model more realistic situations.

3.1. Agents and Tasks

Let $A = \{1, 2, \dots, n\}$ be a set of agents. Agent i has p kinds of resources that are described by a vector $\mathbf{h}_i = (h_i^1, \dots, h_i^p)$, where h_i^k represents the amount of the k -th resource and is a non-negative integer. We assume that task $T = \{S_T, u_T\}$ consists of a set of subtasks $S_T = \{s_1, \dots, s_l\}$ and an associated utility $u_T \geq 0$. The execution of T means the executions of all subtasks $s_j \in S_T$ (this is also denoted by $s_j \in T$ for descriptive purposes). When T is successfully executed, the agents doing T will receive a utility, u_T . Subtask s_j requires various resources, and these resources are denoted by a vector $\mathbf{r}_{s_j} = (r_{s_j}^1, r_{s_j}^2, \dots, r_{s_j}^p)$, where $r_{s_j}^k$ is a non-negative integer and at least, $\exists k_0$ s.t. $r_{s_j}^{k_0} > 0$, expressing the required amount of k -th resource. For simplicity, we assume that u_T is the sum of all resources required by T ; thus, $u_T = \sum_{k=1}^p r_T^k$, where $r_T^k = \sum_{s \in T} r_s^k$. We also denote $u_s = \sum_{k=1}^p r_s^k$. Our method is applicable when u_T is not the sum of the required resources.

For $s \in T$, agent i can execute s when

$$\forall k, h_i^k \geq r_s^k. \quad (1)$$

Agent i can execute multiple subtasks simultaneously if it has more resources than the sum of the required resources of subtasks; i.e., if

$$\forall k, h_i^k \geq \sum_{s \in S} r_s^k, \quad (2)$$

holds, i can execute all the subtasks in $S \subset T$.

A team for executing task T is defined as (G, σ, T) , where G is the set of agents. For each subtask s in T , function $\sigma(s) = i \in G$ expresses the assignment of s i.e., s is allocated to agent i . We say that a team for executing T has successfully been formed when the equation

$$h_i^k \geq \sum_{s \in \sigma^{-1}(i)} r_s^k \quad (3)$$

holds for $1 \leq \forall k \leq p$. When $\sigma(T) = G$, we say the team (G, σ, T) is *lean*. We assume that no agent can simultaneously belong to multiple teams. Thus, while multiple subtasks can be executed if Condition (2) holds, they must be elements of the same task.

Once the team for T has been successfully formed (i.e., the agents in the team execute the assigned subtasks in T), the leader, as the representative of the team, receives the sum of the utilities for all tasks, u_T . The leader collects a certain ratio of the utility u_T in return for acting as the leader, divides up the rest according to the resource amounts required for each subtask, and allocates it to the other members as returns. The ratio of the return to the processed subtask is learned by each agent so that they can expect such future received utilities when they join the leader's team again. The purpose of our research is to increase the sum of received utilities in all the agents by improving the success rate of team formation by self-interested agents.

3.2. Team Formation

Let $Q = \langle T_1, T_2, \dots \rangle$ be the queue of tasks requested by the environment. An (idle) agent peeks at the tasks, starting at the head of the queue, and selects the first unmarked task T . It then marks T as “peeked.”²

First, an agent determines its role, *leader* or *member* in a certain team by referring to the content of marked task, T and the teams that it is solicited to join. How agents determine their roles will be explained in Section 4.3. An agent that decides to act as a leader voluntarily attempts to form a team for T and removes T from Q . An agent that decides to act as a member selects one of the solicited teams. T is then unmarked.

Leader $l \in A$ for carrying out T first selects as many subtasks in T as possible:

$$\arg \max_{S \in 2_l^T} \sum_{1 \leq k \leq p} \sum_{s \in S} r_s^k \quad (4)$$

where $2_l^T = \{s \in S^T : s \text{satisfies Condition (2)}\}$ and 2^T is the power set of T . It then selects the members to allocate the remaining subtasks to, by using the strategy described in Section 4. The set of members together with l is called a *pre-team* and denoted by

²This marking process is to prevent multiple teams forming for a single task.

G^p . Next, l sends *team solicitation messages* with a number of subtasks to the selected members and waits for their responses. Suppose that G^0 is the set of l and the agents that sent acceptance messages to l . l assigns the subtasks to the members of G^0 (this is also explained in Section 4). This assignment is denoted by σ , and the set of members that are assigned at least one subtask is denoted by $G(\subset G^0)$. If (G, σ, T) satisfies Formula (3), l informs the members in (G, σ, T) that a *team* has successfully been formed. Note that this method of selecting members generates a lean team by excluding redundant members. l sends regret messages to the members that have been assigned no tasks, and these agents may seek to join another team in the next round. After that, all the agents in (G, σ, T) execute the subtasks in T , and return to an idle state. If l cannot find a team satisfying Formula (3), the formation of a team for T fails and l sends *team abort messages* to the members in G^0 . The agents in G^0 then enter an idle state. Note that this execution may change the order of tasks.

Agent i that has decided to act as a member looks at the team solicitation messages that have been received and selects one message whose requested subtasks are executable in i that is, Condition (2) is satisfied. How this message is selected will be discussed in the next section. It then sends an acceptance message to join a team and sends *member declination messages* to the other leaders. If it does not receive any team solicitation messages or receives a team abort message from the leader, it returns to an idle state.

We introduce a unit of time called a *tick*, which is the minimal simulation time. A number of tasks in \mathcal{T} is generated and added to Q every tick. The number of added tasks is called the *workload* and is denoted by λ .

4. Learning for Team Formation with Resource Estimation

We explain the proposed learning method, which is the parameter learning for team formation described in [3] together with the resource estimation of other agents. First, we briefly describe the parameters used in [3]. Then, we discuss how the resource estimation method is incorporated in the parameter learning.

Three parameters, the *degree of greediness* (DG), the *expected ratio of team solicitation acceptance* (ETSA), and the *expected member utility division ratio* (EMU) are defined in order to calculate dividends for team members and to predict the received utilities if the team were to be successfully formed. Each agent individually learns the values of these parameters through experience. Because we assumed that all agents, both the leaders and members, are self-interested, they would like to take actions to maximize the expected utilities and reduce the team formation failures.

4.1. Degree of Greediness

The *degree of greediness* (DG), g_i ($0 \leq g_i \leq 1$), of a leader agent i is the ratio of the number of utilities that i collects in return for acting as a leader if it succeeds in forming a team (G, σ, T) . Therefore, i collects a utility $u_i^{leader} = u_T \times g_i$. The remaining utilities are allocated according to the resources required for the subtasks among the other members. That is, agent j receives

$$u_j^{member} = (u_T - u^{leader}) \times \frac{\sum_{t \in \sigma^{-1}(j)} u_t}{\sum_{s \in T \setminus \sigma^{-1}(i)} u_s} \quad (5)$$

for $j \in G \setminus \{i\}$.

If g_i is large, the members of the teams formed by leader i cannot expect more utility. They are likely to decline team solicitation messages from i , thus making it difficult for i to act as a leader. To adaptively learn this parameter, i updates the value of g_i , depending on the success or failure of a team being formed, by using:

$$g_i = \alpha_g \times \delta_{success} + (1 - \alpha_g) \times g_i, \quad (6)$$

where $\delta_{success} = 1$ if a team is successfully formed and $\delta_{success} = 0$, otherwise. The parameter α_g is the learning rate of DG, where $0 \leq \alpha_g \leq 1$.

4.2. Expected Member Utility Division Ratio

For agents i and j ($i \neq j$), the *expected member utility division ratio* (EMU) of i for j , $d_{i,j}$, expresses the ratio of the expected utilities received from j to the utility of the subtasks allocated to i when i has accepted a team solicitation message sent by leader j . This parameter is used to determine whether agent i should act as a leader or a member and which team solicitation message it should accept if i acts as a member. The agent uses the ε -greedy strategy as follows, to select a solicitation message. Let $M_o = \{m_1, \dots, m_q\}$ be the set of received team solicitation messages and $l(m_k)$ be the leader that sent message m_k to i . $S(m_k)$ for $m_k \in M_o$ is the set of subtasks that were proposed by $l(m_k)$ to be executed in m_k . Let $M = \{m \in M_o : S(m) \text{ is executable in } i\}$. Agent i selects one of the received messages \tilde{m} with the probability $1 - \varepsilon$:

$$\tilde{m} = \arg \max_{m \in M} \sum_{s \in S(m)} u_s \times d_{i,l(m)}, \quad (7)$$

where $S(m)$ for team solicitation message m is the set of subtasks that $l(m)$ proposed to be executed in m . Agent i also randomly selects one message from M with probability ε .

Once team (G, σ, T) is formed, leader j of this team receives a utility u_T . Leader j then distributes it to the team members (including i), as described in Section 4.1. i then updates the value of $d_{i,j}$ according to the utilities, U , received from j :

$$d_{i,j} = \alpha_d \times \frac{U}{\sum_{s \in S(\tilde{m})} u_s} + (1 - \alpha_d) \times d_{i,j}, \quad (8)$$

where \tilde{m} is the selected team solicitation message. α_d is the rate of learning for EMU, and $0 \leq \alpha_d \leq 1$. Note that if the team formation fails or i is not selected as a member of (G, σ, T) , U is 0. Hence, this parameter partly reflects the team formation success ratio. It is obvious that $d_{i,j} \geq 0$.

4.3. Role Selection based on DF and EMU

Agents decide their roles using DF and EMU parameters. Agent i in an idle state looks for an unmarked task T , starting from the head of queue Q , as described in Section 3.2. It calculates the expected utilities of E_i^{leader} when it acts as a leader of T and E_i^{member} as a member of the proposed team:

$$E_i^{\text{leader}} = \sum_{s \in T} u_s \times g_i \quad \text{and} \quad E_i^{\text{member}} = \sum_{s \in S(\tilde{m})} u_s \times d_{i,l(\tilde{m})}, \quad (9)$$

where \tilde{m} is the selected team solicitation message. i acts as a leader if $E_i^{\text{leader}} \geq E_i^{\text{member}}$. Otherwise, it acts as a member. We assume $E_i^{\text{member}} = 0$ if $M = \emptyset$. If $Q = \emptyset$ or i cannot find any unmarked task, i tries to act as a member. Moreover, if i has received no team solicitation messages, it returns to an idle state. Note that in an extreme case in which all agents decide to act as leaders, their team formations fail. Thus, their DGs lower and some of them are likely to act as members. Of course, this situation may occur only when many tasks are requested in the very early period of learning.

4.4. Expected Ratio of Acceptance of Team Solicitation

For agent i , K_i is defined as the set of agents that i knows, where $i \in K_i$. We assume that i can send solicitation messages only to agents in K_i . Then, for $i \in A$ and $j \in K_i \setminus \{i\}$, the parameter, $e_{i,j}$ called the *expected ratio of team solicitation acceptance* (ETSA) of i to j , expresses the degree of acceptance when i sends a team solicitation message to j . This parameter is used when leader i selects the members of the pre-team, G^p with the resource estimation method and is also used to define the assignment function, σ , according to the value of $e_{i,*}$. The concept behind using ETSA is that agents contributing to past team formations should be selected more often as pre-team members and allocated high-utility tasks if possible.

Agent i updates the ETSA of $j (\neq i)$ depending on whether j has accepted team solicitation messages by using:

$$e_{i,j} = \alpha_e \times \delta_{\text{accept}}^j + (1 - \alpha_e) \times e_{i,j}, \quad (10)$$

where $\delta_{\text{accept}}^j = 1$ if j has accepted a team solicitation and $\delta_{\text{accept}}^j = 0$ otherwise. α_e is the rate of learning for the ETSA, $0 \leq \alpha_e \leq 1$.

4.5. Resource Estimation and Member Selection

4.5.1. Resource Estimation:

Agent, $i \in A$, has the estimated resources \tilde{h}_j^k of agent $j (\neq i)$ for $1 \leq k \leq p$. First, the initial values of these estimated values are set to 0. Suppose that i acts as a leader and sends to j a team solicitation message with subtask s whose resources are (r_s^1, \dots, r_s^p) . If i receives the acceptance message from j , the estimated values are redefined as $\tilde{h}_j^k = \max(\tilde{h}_j^k, r_s^k)$, since s is executable in j . However, if i receives the declination message from j , i keeps the estimated values unchanged, because there are possibly various reasons for declining an invitation.

To learn the estimated resource values, i has a certain strategy for selecting pre-team members. This strategy determines whether agents exploit the resource values that have been estimated or continue to estimate the resources of other agents. The next section explains how the members of a pre-team or team are selected.

Require: T : task, i : leader agent, L : number of solicitation messages

$S \subset T$: Set of subtasks allocated to i (as described in Section 3.2).

$T = T \setminus S$; $\sigma^p = \emptyset$; $G^p = \emptyset$

T is sorted by the numbers of required resources, u_s .

for all $s \in T$ from the first element, **do**

$l(s) = 0$; $K = K_i \setminus \{i\}$

for $j \in K$: selected according to $e_{i,j}$ **do**

if $\tilde{h}_j^k \geq r_s^k$ for $\forall k$ **then**

$\sigma^p = \sigma^p \cup \{(s, j)\}$ // Allocate s to j

$G^p = G^p \cup \{j\}$

$\tilde{h}_j^k = \tilde{h}_j^k - r_s^k$ for $\forall k$

$l(s) = l(s) + 1$; $K = K \setminus \{j\}$

if $l(s) \geq L$ **then**

 break. // Exit inner for-loop.

end if

end if

end for

end for

for all $s \in T$ such that $l(s) < L$ **do**

 Select $L - l(s)$ agents $\{a'_1, \dots, a'_{L-l(s)}\}$ from $K_i \setminus G^p$, randomly

$\sigma^p = \sigma^p \cup \{(s, a'_1), \dots, (s, a'_{L-l(s)})\}$

$G^p = G^p \cup \{a'_1, \dots, a'_{L-l(s)}\}$

end for

Figure 1. Calculation of pre-team G^p and their assigned subtasks in T .

4.5.2. Determination of Pre-Team Members:

This is the part of the proposed method that is quite different from the previous one. Leader i uses the ε -greedy strategy to determine the members of the pre-team G^p for T . G^p is initially set to $\{i\}$, and i determines the set of the subtasks, S , that i will do itself, as described in Section 3.2.

First, leader i randomly selects L agents members of G^p for each $s \in T \setminus S$, where L is a positive integer denoting the number of solicitation messages to send for each s_k .

Otherwise, i decides G^p , with probability $1 - \varepsilon$, as follows (see also Figure 1). The elements of $T \setminus S$ are sorted according to the required utility; this is denoted by $\langle s_1, \dots, s_m \rangle$, where $u_{s_1} \geq u_{s_2} \geq \dots \geq u_{s_m}$. Then, the other members of pre-team G^p for T are determined according to this order. For the l -th subtask $s_l \in T \setminus S$, i selects the first agent, a_1 , from $K_i \setminus \{i\}$ according to the value of ETSA in i . If i estimates that a_1 has enough resources to execute s_k according to the estimated resources \tilde{h}_1^k ($1 \leq k \leq p$) of a_1 , it selects a_1 for s_l and $G^p = G^p \cup \{a_1\}$. Note that if a_1 has already been selected for another subtask, say \tilde{s} , and if it has sufficient resources to carry out both \tilde{s} and s_l , a_1 is selected for s_l as well as \tilde{s} . Then, i moves on to check the next agent, $a_2 \in K_i \setminus \{i, a_1\}$. If a_1 does not have enough (remaining) resources for s_l , i moves on to check a_2 . This process is iterated until i has selected L agents for s_l ($1 \leq l \leq m$). If i cannot find L agents to allocate s_l , it randomly selects one from $K_i \setminus \{i\}$ so that L agents will be assigned to task s_l . Then, it selects s_{l+1} and starts the selection process for s_{l+1} from $K_i \setminus \{i\}$ in the same way.

After that, the team solicitation messages containing the selected subtasks are sent to the members in G^P . The subtask and selected agent pairs are denoted by σ^P in Figure 1. Note that L indicates the redundancy of team solicitation messages and, of course, a larger L will reduce the chance of a team formation failure, though it generates more messages. The solicitation message is broadcast when $L = |A|$, as in the contract net protocol. Note that L was set to 2 in our experiments; this setting dramatically reduced the number of messages sent during team formation process and is one of the characteristics of the proposed method.

4.5.3. Determination of Team Member and Assignment Function:

Let G^0 be the set of agents that have accepted team solicitation messages. First, leader agent i revises the estimated resources of agents in G^0 according to their responses. Then, i defines σ by allocating the tasks as follows: For $s \in T \setminus S$, if there are agents in G^0 for s , one agent is selected using the ε -greedy strategy by checking values of ETSA and is assigned s . After that, if there is a subtask s' that no agent has accepted yet, i tries to allocate s' to the agents in G^0 that can afford to do it. If i cannot allocate s' to the agents in G^0 , it decides that the current process of team formation has failed. When all subtasks are allocated, we can say that team (G, σ, T) has been successfully formed, where G is the set of agents to which the above process at least one task and σ comprises as the assignments of in this process.

5. Experimental Evaluation

5.1. Experimental Settings

We experimentally measured how much the total received utility was increased by the proposed method and compared it with the increase brought by the previous method proposed in Ref. [3]. The previous method is called the parameter learning (PL) method. In this experiment, fifty agents were generated, i.e., $|A| = 50$. We set $L = 2$ and $K_i = A \forall i \in A$. The initial values of the learning parameters are in Table 1. The number of resource types was two ($p = 2$). The amount of each resource in agent i , $h_i^k (k = 1, 2)$ was a randomly selected integer between 3 and 12 ($3 \leq h_i^k \leq 12$). A task consisted of 3 to 7 subtasks, and the required resource for each subtask s , r_s^k , was also randomly set as a positive integer between 1 and 8. Two tasks on average were generated every tick according to a Poisson distribution ($\lambda = 2$) and were queued. We recorded the number of executed tasks every 50 ticks, from the 1 to 30000 ticks. The data below are the average values for a hundred independent trials based on different random seeds.

Table 1. Initial Values for Learning Parameters.

Parameters	Initial values	Parameters	Initial values
DG ($g_i \forall i \in A$)	Random from [0, 1]	α_g	0.1
ETSA ($e_{i,j} \forall i, j \in A$)	0.5	α_e	0.05
EMU ($d_{i,j} \forall i, j \in A$)	0.5	α_d	0.05

We also compared our results with those obtained by two other methods. One method was CNP; CNP is a market-based task allocation method for cooperative agents. In CNP, no manager knows the resources/capabilities of other agents like ours, so it is a good benchmark for evaluation. The crucial difference from our method is that CNP generates many messages in the announcement and bidding processes.³ We conducted the experiments when the numbers of managers are 1, 2, 3, 4, 5, 10, because they affect the overall performance. These experiments are denoted by CNP n , where n is the number of managers. The second method is a non-learning (NL) method in which agents select their role randomly and for each subtask $s \in T$, the leader agent l randomly selects L among the set of members that satisfies condition (1) for s . Because the NL method does not learn any of the parameters described in this paper, its selection of roles and members was random. However, since the agents are self-interested, they select the team solicitation message whose required resources are maximal. Furthermore, in the NL method as in the PL method, all agents know the resources of the other agents, and leaders can make a pre-team with accurate information. This assumption is unrealistic to apply to actual systems, and it is the difference of NL and PL methods from CNP and the proposed method. We did not describe the result by [10], because it is known that its performance was lower than the PL method [3].

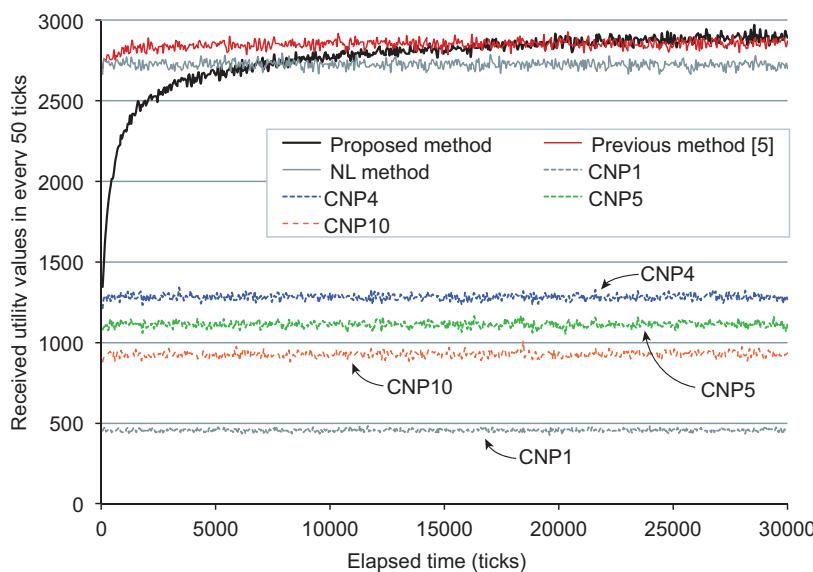


Figure 2. Transition in numbers of received utilities.

³In CNP, announcements are broadcast. This does not mean there is a single broadcast message in the datalink layer, because CNP is an application-layer protocol. Thus, broadcasts are usually realized by sending many unicast messages to the target agents.

5.2. Experimental Results and Discussion

Figure 2 shows the change in the received utility as a result of executing tasks (as the result of successful team formation) over time. It is clear that the proposed method performed as well as our previous method and outperformed the NL method and the CNPn. During the period from 29000 to 30000 ticks, the proposed method's cumulative utility was approximately 0.3% higher than that of PL, 5.8% higher than that of NL, and 125.2% higher than that of CNP4 that exhibited the best among CNPn.

These results are somewhat counter-intuitive: the proposed method must be approximately 5% lower than the previous method, because the ϵ -greedy strategy randomly select the pre-team members for learning. Of course this is necessary to estimate agents' resources, but when its members are random, the team formation hardly succeeds. So, for example, if agents make ϵ gradually smaller over time such as the Boltzmann exploration strategy, we may slightly improve the performance in the proposed method. We need more additional experiments, but we believe that this is also the result of the small randomness in the ϵ -greedy strategy when a leader selects pre-team members. This strategy increases the chances to send team solicitation messages to agents whose ETSA parameter is low. In PL method, the ETSA values sometimes decrease; for example, when leader l sends a solicitation message to agent i whose ETSA value $e_{l,i}$ is relatively high, i cannot accept it if it is already a member of another team. So it becomes a little more difficult for l to select not only pre-team members, but also team members. This also decreases the value of EMU in i . However, in the proposed method, agents occasionally recover the values of $e_{l,i}$ because of their random selection with probability ϵ .

The difference in performance between NL and PL indicates the effect of the parameter learning proposed in [3]. We think that learning the resources of other agents can develop the potential capabilities of parameter learning. Note that if the number of solicitation messages, L , is larger in the proposed, PL and NL methods, it is obvious that the success ratio of team formation will increase and the learning speed will also increase. However, we would like to investigate the effect of learning with the resource estimation on received utility by making the number of L minimal in this experiment.

Surprisingly, Figure 2 also shows that CNP could earn the fewest utilities among the four methods. We believe that this poor result was caused by the *self-interest* of the agents. In CNP, a task announcement, corresponding to a team solicitation message, is broadcast by managers, and contractor agents select one announced task by referring to their own local viewpoint. Since they are self-interested, they are likely to select specific tasks that they could execute with more resources in order to receive more utilities. This may have resulted in a concentration of bid messages. Furthermore, agents could not belong to more than one team at a time, and thus many team formation failures occurred. This discussion also suggest the numbers of managers strongly affect the performance in CNPn; actually the performance is the best when the number of manager is four. Note that we did not show the results of CNP2 and CNP3 but the results of CNP2 and CNP3 were almost identical to those of CNP10 and CNP5, respectively. Thus, this means that there is the trade off between concurrency and concentration of task allocations with CNP.

6. Conclusion

To achieve efficient team formations without the assumption that agents have information about the resources/capabilities of other agents, we proposed the parameter learning method with resource estimation to form teams for given tasks by self-interested agents. In this method, all agents learn three parameters related to team formation from the viewpoint of both leaders and members. They also learn the resources of other agents according to the responses to team formation solicitations. We then experimentally evaluated the proposed method by comparing it with CNP, a non-learning method, and the PL method which assumes information about resources in agents. Our learning method increased the received utility in comparison with the PL method, although the PL method required more information about other agents to decide on team members.

We would like to extend our learning method so that it can be applied to large-scale multi-agent systems in which hundreds or thousands of agents work together[11]. For such an extension, we think that we will have to revise the set of K_i based on past experience.

Acknowledgement: This work was, in part, supported by KAKENHI (22300056).

References

- [1] J. M. Corchado, D. I. Tapia, J. Bajo: A Multi-Agent Architecture for Distributed Services and Applications, *International Journal of Innovative Computing, Information and Control* **8**(4) (2012), 2453–2476.
- [2] M. N. Huhns, et al.: Research directions for service-oriented multiagent systems. *IEEE Internet Computing* **9** (2005), 65–70.
- [3] D. Hamada, T. Sugawara: Deciding roles for efficient team formation by parameter learning, *Proceedings of the 6th KES international conference on Agent and Multi-Agent Systems: technologies and applications (KES-AMSTA'12)* (2012), Springer-Verlag, 544–553.
- [4] R. G. Smith: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers* **C-29**(12) (1980), 1104–1113.
- [5] Fipa contract net interaction protocol specification. <http://www.fipa.org/specs/fipa00029/SC00029H.html/> (2002).
- [6] O. Shehory, S. Kraus: Feasible formation of coalitions among autonomous agents in nonsuperadditive environments. *Computational Intelligence* **15**(3) (1999), 218–251.
- [7] J. Yang, Z. Luo: Coalition formation mechanism in multi-agent systems based on genetic algorithms, *Appl. Soft Comput.* **7**(2) (2007), 561–568.
- [8] S. Abdallah, V. Lesser: Organization-based cooperative coalition formation, *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (2004), 162–168.
- [9] R. Katayanagi, T. Sugawara: Efficient team formation based on learning and reorganization and influence of communication delay, *Proceedings of 11th IEEE International Conference on Computer and Information Technology* (2011), 563–570.
- [10] T. Genin, S. Aknine: Coalition formation strategies for self-interested agents in task oriented domains, *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (2010), 205–212.
- [11] T. Sugawara, T. Hirotsu, S. Kurihara, K. Fukuda: Performance Variation Due to Interference Among a Large Number of Self-Interested Agents. *Proceedings of 2007 IEEE Congress on Evolutionary Computation* (2007), 766–773.

Incorporating Explicit Coordination Mechanisms by Agents to Obtain Green Waves

Antonio de Abreu BATISTA JÚNIOR ^{a,1} and Luciano Reis COUTINHO ^{a,2}

^aComputer Science Department the Federal University of Maranhão, São Luís, Brazil

Abstract. This paper describes a multi-agent system (MAS) acting on an arterial road network, where each intersection is controlled by an agent. The agents are concerned with the efficient control of their intersection. In order to improve the overall performance of the system we propose a model of explicit coordination that directs the behavior of agents for the formation of green waves on the artery, in addition to maintaining the autonomy of each agent. The model is tested in simulation and compared with the traditional approach of synchronization of traffic lights. The results obtained in simulation overcome the traditional model by representing a more realistic model of traffic.

Keywords. intelligent traffic control, simulation, multi-agent coordination, BDI reasoning

1. Introduction

Traffic control in the urban areas is becoming increasing complex with the exponential growth in vehicle count. Expansion of the road network to accommodate the increased vehicle count is an expensive solution and harmful to the environment [1]. As an alternative the expansion of the road network can improve the utilisation of the existing infrastructure [1] [2]. The coordination of successive signals along an artery has the potential to improve the overall performance of a road network. However, providing a closed-loop control strategy that is adaptive to the fluctuations in traffic conditions and that take into account the dependencies between neighboring intersections is an extremely challenging distributed control problem.

Multi-agent systems have shown great potential for self-scheduling traffic signal control in the stochastic traffic environment [1] [3], the main advantage of which is the ability to provide a decentralized functional and spatial approach [4]. Such systems are

¹Corresponding author: Antonio de Abreu Batista Júnior is a master student in computer science at the Federal University of Maranhão, São Luís, Brazil; E-mail: junior2004@pop.com.br

²Luciano Reis Coutinho is Brasilian research chair in multi-agent system and head of the Department of Computer Science at the Federal University of Maranhão, São Luís, Brazil; E-mail: lrc@deinf.ufma.br

composed of multiple autonomous agents capable of perceiving the environment and deciding for themselves their own course of action in order to achieve a common goal.

In multiagent context, interaction between agents is the main feature. The most of the efforts in developing multi-agent systems is directed to the interaction between the agents and the rules that focus on these interactions, since the precise coordination and management activities among agents are essential to the efficient functioning these systems. The coordination among agents can be achieved implicitly by exploiting the structural information in the environment where they operate or explicitly through the direct communication between agents [4].

This paper describes a multi-agent system acting on an arterial road network, where each intersection is controlled by an agent. In order to improve the overall performance of the system we propose a model of explicit coordination that directs the behavior of agents for the formation of green waves on the artery, in addition to maintaining the autonomy of each agent. We present simulation results on one arterial network with dynamic vehicle flows that demonstrate the performance benefit of our self-scheduling approach.

This paper is organized as follows. In section 2 we present some works related to coordination of traffic signals. In Sections 3 and 4, the problem formulation and proposed solution are presented. Next, sections 5 and 6 discuss the details about the implementation of the simulation model and simulation results respectively. Finally, Section 7 concludes the paper and outlines the future research directions.

2. Related Works

In this section we discuss some works that taken into account the dependencies between neighboring intersections. At the end we point the benefits of our approach.

In [1], a self-scheduling approach to solve the traffic signal control problem, where each intersection is controlled by a self-interested agent operating with a limited (fixed horizon) view of incoming traffic. Central to their approach is an aggregate representation of traffic flows as critical clusters of anticipated queues and platoons. These aggregate patterns provide a basis for real-time signal control policies that incorporate greater look-ahead and extend the queue clearing strategies utilized by prior work [5]. They design two platoon-based policies aimed at deciding whether or not to extend a phase through idle periods of no traffic in order to serve future traffic, with the goal of promoting implicit coordination between successive signals and the establishment of green waves, and ultimately improving overall performance.

In [6], it is proposed a multi-agent traffic light control framework that combines both indirect and direct coordination. Reaction to dynamic traffic flow is attained by indirect coordination, and green-wave formation is attained by direct coordination. In normal daily traffic flow, each agent functions based on the indirect coordination mode. However, when the traffic flow balance collapses near certain agents, the agents change their coordination mode to direct coordination mode to form a green-wave control formation.

In [7], it is presented an approach where each traffic light behaves like a social insect, having coordinated signals plan as tasks to be performed. The model uses a restricted communication mechanism and coordinated groups are formed in a dynamic way. the approach intends to combine the advantages of decentralization via swarm intelligence

and dynamic group formation. The main advantage this approach is the adaptation to changes in the traffic. Changes are perceived and the agents react to these changes is a fast and independent form, without any hierarchical organization.

In [8], a hierarchical multi-agent system that consists of several locally operating agents each representing an intersection of a traffic system is proposed. Local Traffic Agents (LTAs) are concerned with the optimal performance of their assigned intersection; however, the resulting traffic light patterns may result in the failure of the system when examined at a global level. Therefore, supervision is required and achieved with the use of a Coordinator Traffic Agent (CTA). A CTA provides a means by which the optimal local light pattern can be compared against the global concerns. The pattern can then be slightly modified to accommodate the global environment, while maintaining the local concerns of the intersection. The limitation is the necessity of a centralized architecture where decisions concerning the green time, offset, and cycle time are centrally made by Coordinator Traffic Agent (CTA).

In [9], it is developed a traffic flow model consisting of the following elements: roads, traffic lanes, road intersections, traffic lights and vehicle volumes. To ensure the model achieved its expected results an ant algorithm was applied whereby vehicles moving on the roads in the model left pheromone depending on how long they would wait to cross a particular intersection. For congested roads with poorly coordinated traffic light sequences, an appropriate level of deposited pheromone would encourage traffic to be directed in an appropriate direction. This relationship between waiting time and pheromone concentration also worked as a signal initiating changes in traffic light sequences at the intersections so that vehicles could more rapidly exit from the congested area.

The previous works are all attractive despite some undesirable functional characteristics such as centralization and intuitive decision-support. In our work we propose a multi-agent system to act on an artery, where decision-support each agent is based on human practical reasoning. This model has been used successfully in situations where the modeling of human reasoning is required. Next we define the limitations imposed by the conventional approach of synchronizing signals on an artery.

3. Problem Formulation

The coordination of successive signals along an artery has the potential to improve the overall performance of a road network. It allows vehicles, traveling with a certain constant average speed, travel freely on an artery. However, the traditional approach of fixed synchronization can not handle variable traffic patterns by simplifying the model. Allied to this, in some cities, cross roads have also become important due to the saturation of arterial roads. In short, the lack of local adaptation of this approach causes delays at the cross roads. A Figure 1 illustrates this scenario.

In general this is the problem of determining in real time which plan is more appropriate for each signal in order to keep a road network with better mobility.

Next we define our adaptive strategy.

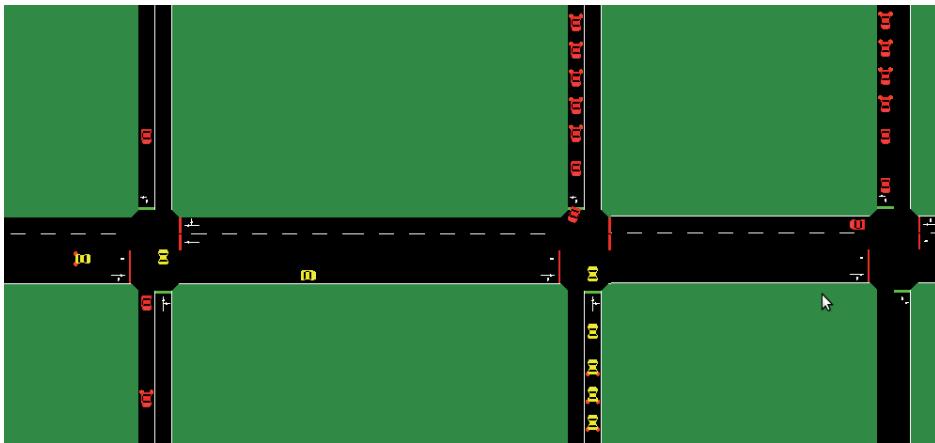


Figure 1. Example of green wave causing congestion on cross roads.

4. Proposed Model Self-scheduling

The definitions that follow describe a BDI agent to act on an artery and the way they interact to achieve coordination and local adaptation.

4.1. BDI Agent

A BDI agent is a program formed by a set of beliefs B , a set of desires D and a set of intentions I . The beliefs of an agent are used to determine which pre-conditions of plans in the agent's plan library are satisfied. In this work, B_j represents beliefs of agent j about the intersection j that it controls.

The beliefs of the agent j about the intersection j are a set of assignments of truth values to propositions p about it, defined as follows:

$$B_j = \langle cp_j, Sat^1_j, \dots, Sat^i_j \rangle$$

where, cp_j is a of plans of the agent j 's plan library and indicates which the plan currently implemented at the intersection j . The Sat^i_j is the degree of saturation of each critical link of each phase i of the intersection j . The critical link is the most loaded link of each phase i .

$$Sat^i_j = \frac{VecIn}{VecOut} \quad (1)$$

$VecIn$ is the number of vehicles that wishes to pass the retention area of the intersection j and $VecOut$ is the number of vehicles that can get through the retention area of the intersection j both measured over a certain period of time.

In practical terms, the degree of saturation of a link reflects their level of loading. For instance, If the degree of saturation of one link is equal to 50% $Sat^i_j = 0.5$, this means that could have passed two times the number of vehicles during the green time adopted. Another example, If the degree of saturation of one link is equal to 150% $Sat^i_j = 1.5$ means that two thirds of vehicles could pass the retention area while one-third was retained for the next cycle.

The desires of an agent are the post-conditions of plans from the agent's plan library. The desires of an agent j are a set of pairs $\langle d, Cond_{d_j} \rangle$ such that, d is a desire and $Cond_{d_j}$ is an assignment to a proposition p . The interpretation is that j believes that the desire d is satisfied if $Cond_{d_j}$ is satisfied with respect to B_j .

Finally, the intentions of an agent represent desires that it has committed to achieving. BDI agent to act on an artery selects a desire d by means of the protocol described in the next subsection.

4.2. Interaction Protocol

The protocol defines two roles: base agent (*BA*) and common agent (*CA*). The *BA* should control the first of n consecutive adjacent intersections on the artery. Figure 2 illustrates the exchange of messages between agents:

1. The *BA* should select a desire $\langle d, Cond_{d_j} \rangle$ through its B_j and inform it the others community members (*CA*). A desire can be translated as a plan and a condition in which the plan should be adopted.
2. Each participant (*CA*) to become aware of the new desire, sends a message to agent *BA* saying that commits to reach it.

The synchronization on the artery is achieved when each agent (*CA*) reaches its goals (the desire of the base agent) since the time lag between the start of green time of each plan (offset) was previously configured for this purpose.

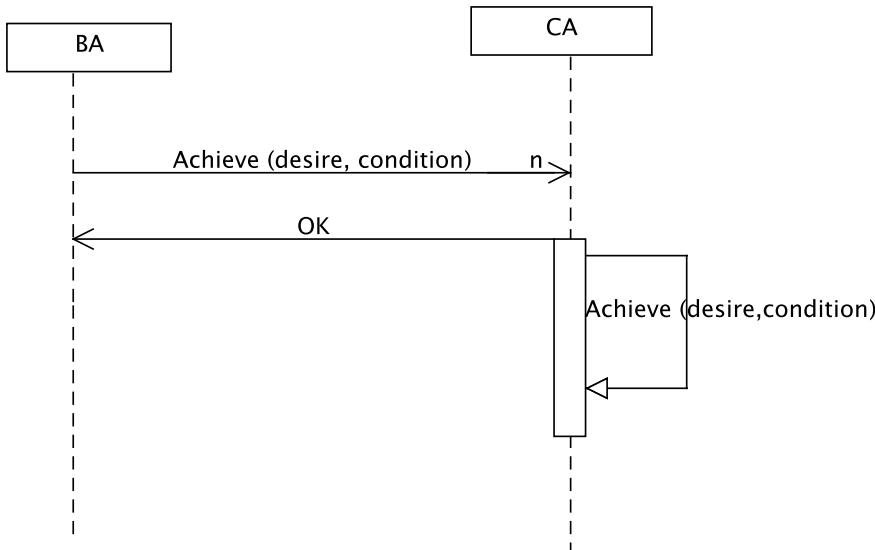


Figure 2. Interaction protocol for creating green wave.

4.3. Autonomy of Agents.

Each agent is equipped with a library of pre-compiled plans. For all desire of an agent j , $\langle d, Cond_{d_j} \rangle$, there exists a pre-compiled plan to achieve it. An agent selects a plan based on the idea that cycle times larger can pass more vehicles and shorter cycle times reduce the waiting time at the signal. Each agent uses the degree of saturation of each critical link to help it in this task.

An agent j performs the desire $\langle d, Cond_{d_j} \rangle$ if $Cond_{d_j}$ to a proposition p is satisfied with respect to B_j . The local adaptation is achieved by autonomous agents. They need to maintain an appropriate state of affairs (traffic situation which must be maintained for using the plan informed by base agent). Thus, agents keep in mind the desire to implement the target plan and to satisfy their wills they modify the environmental conditions to which the preconditions of the target plan are met and finally they can adopt the plan. Once they have achieved your desires, they continually observe the post-conditions of the plan in order to take actions to correct it if the post-conditions of the plan do not agree with their beliefs.

5. Implementation Model

Next we discuss the implementation of the model.

5.1. Building the Simulation Environment

The interpreter Jason [10] was used to create the agents and their environment. Jason allows developing cognitive agents based on the BDI model programmed in AgentSpeak (L), including communication between agents based on the theory of speech acts. The agents act in the environment of traffic simulator SUMO [11] (Simulation of Urban Mobility) an open-source microscopic road traffic simulator. Each agent can percept and modify the current condition of traffic at the intersection under its responsibility. When an agent decides to adopt a program semaphore, a process is executed to modify the current semaphore program in the simulator environment to reflect the will of the agent. Likewise, the perception of agents is obtained through sensors throughout the simulator environment, which capture aspects of the properties current traffic situation.

In order to integrate the Jason to SUMO, Figure 3, we use the XTRACI³, a Java API which implements some basic interfaces to provide remote access to control traffic signal in real time in SUMO. Finally, we build our system multiagent putting artificial agents (AgentSpeak) to operate in the simulator environment.

5.2. Abstract Simulation Scenario

The proposed multi-agent approach was tested on a simulated arterial network of 5 intersections. The simulated network is the highly congested section of the Guajaráas avenue in São Luís. The network serves as an ideal test bed because of the geometry and heterogeneity in the classification of links (major and minor roads with varying speed limits). The network, Figure 4, is simulated using SUMO.

³<http://www.cs.cmu.edu/~xfxie/download/xtraci1.0.tar.gz>

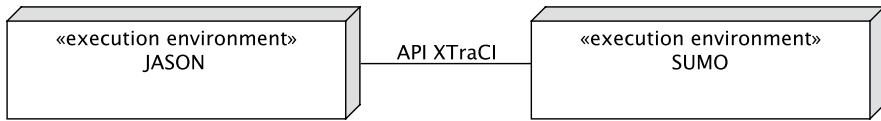


Figure 3. Integrating Jason to SUMO.

We create the routes with random traffic of 4 different kinds of vehicles, see Table 1. The vehicles leave the source according to a Poisson process approximated here by a binomial distribution. So a parameter $p=1/40$ in the table of the Figure 4 means that a vehicle is generated every 40 seconds in average at that particular source. This makes the simulation close to reality where the São Luís traffic is unpredictable and heterogeneous in nature.

Table 1. Types of vehicle used in the case study of Sā Luís

Types of vehicle	acceleration	deceleration	length	maximum speed
CarSlow	1.0 m/s^2	6.0 m/s^2	6.0 m	40 km/h
CarMiddle	2.0 m/s^2	8.0 m/s^2	5.0 m	60 km/h
CarFast	3.0 m/s^2	9.0 m/s^2	3.0 m	100 km/h
Bus	0.1 m/s^2	2.0 m/s^2	10.0 m	30 km/h

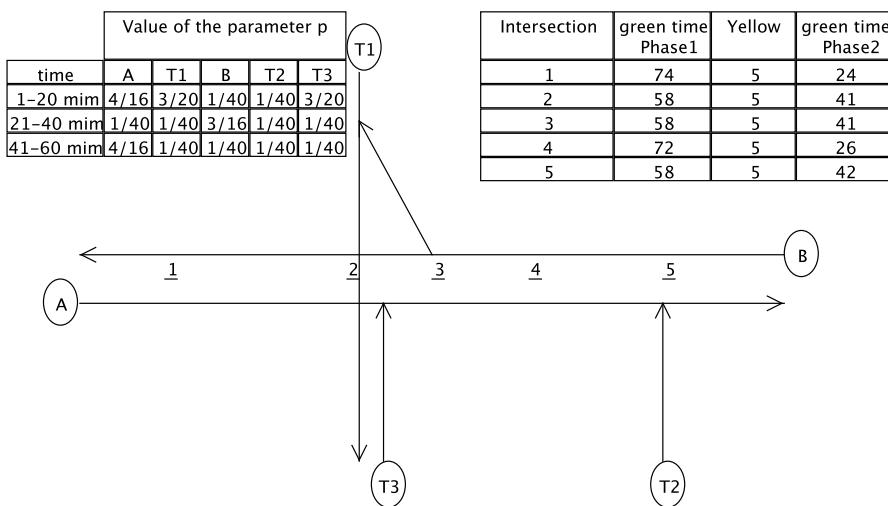


Figure 4. Session simulated Avenue.

6. Simulation Results and Discussions

Experiment aims to compare the proposed solution against the solution currently implemented in the section of the Guajajáras avenue. To measure the performance of both ap-

Table 2. Plan of the agents.

plan	cicle	green time phase1	green time phase2	aim
plan 1	120	80	40	daily plan
plan 2	120	30	90	prioritize the phase 2

proaches, we used the average total delay parameter AD that indicates how good was the choice of plans at each intersection in order to keep a road network with better mobility [2].

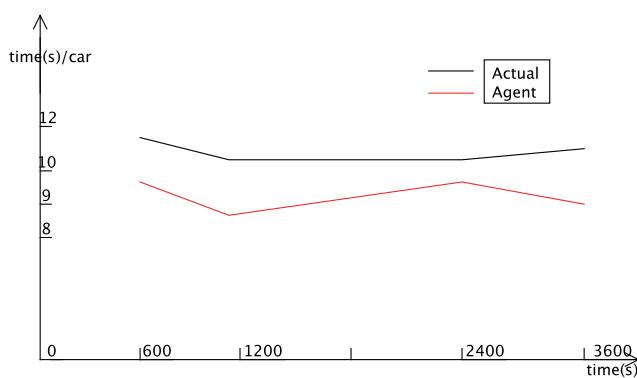
$$AD = \frac{\sum_{i=1}^n T_{Di}}{V} \quad (2)$$

where, n is the number of intersections in the arterial network, T_{Di} is the delay experienced by vehicles at each intersection $i = \{1, 2, \dots, n\}$ and V is the total number of vehicles that entered and left the network during the period measured.

For this simulation each agent was equipped with two planes. Table 2 illustrates these plans.

Ten simulation runs using different random seeds were carried out for each signal control technique compared. Since variance of the outcome of simulation runs was small, average value was taken as the representation of the outcome.

Figure 5 shows comparison of the time average total delay by vehicles in road network using different types of traffic signal control. The proposed multi-agent control shows a 19% improvement in delay in comparison to the solution currently implemented in the section of the Guajajáras avenue. The improvement in performance can be attributed to the ability of organizing agents create smooth flow by using the interaction protocol, which increases the overall view of the agents, and the ability of each to maintain their local concerns do not leave build queues in the cross roads.

**Figure 5.** The evolution of the average total delay AD measured in one hour simulation.

7. Conclusion

In this paper, we proposed a model of explicit coordination that directs the behavior of agents to synchronize themselves, while it is maintained the autonomy of each agent. The practical simulation experiments confirmed the expectations of researchers to achieve an efficient local control through the autonomy of agents and in normal traffic creating green waves on the artery. BDI architecture was used to build rational agents capable of independent and autonomous actions to allow them to decide for themselves given some situation.

And although we achieved good experimental results, they were not better due to simplifications of the model that does not consider the possibility of coordination in different directions. In our future work we intend to extend the protocol to cover the circular traffic flow. we also intend to compare our model with other strategies and analyze the issue of the number of messages exchanged.

References

- [1] X. Xie, G.J. Barlow, S.F. Smith, Z.B. Rubinstein, Platoon-Based Self-Scheduling for Real-Time Traffic Signal Control, *In: IEEE International Conference on Intelligent Transportation Systems (ITSC)* (2011), 879–884.
- [2] P.G. Balagi, X. German, D. Srinivasan, Urban traffic signal control using reinforcement learning agents, *In: IEEE International Conference on Intelligent Transportation Systems (ITSC)* (2010), 177–188.
- [3] S. El-Tantawy, B. Abdulhai, An Agent-Based Learning Towards Decentralized and Coordinated Traffic Signal Control, *In: Annual Conference on Intelligent Transportation Systems* (2010), 19–22.
- [4] A.L.C Bazzan, A Distributed Approach for Coordination of Traffic Signal Agents, *In: Autonomous Agents and Multi-Agent Systems Vol. 10* (2005), 131–164.
- [5] S. Lämmer and D. Helbing, Self-control of traffic lights and vehicle flows in urban road networks, *Journal of Statistical Mechanics: Theory and Experiment* (2008), 4–19.
- [6] T. Shirai, Y. Konaka, J. Yano, S. Nishimura, K. Kagawa, T. Morita, M. Numao and S. Kurihara, Multi-agent traffic light control framework based on direct and indirect coordination, *In: Proceedings of the 7th International Workshop on Agents in Traffic and Transportation* (2012), 9–17.
- [7] D. Oliveira, A.L.C Bazzan, Swarm Intelligence Applied to Traffic Lights Group Formation, *In: VI Encontro Nacional de Inteligência Artificial (ENIA 2007)* (2007), 1003-1012.
- [8] J. France and A. A. Ghorbani, A multiagent system for optimizing urban traffic, *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03) 0-7695-1931-8* (2003).
- [9] D. Król, M. Mroek, Swarm-based Multi-agent Simulation: a Case Study of Urban Traffic Flow in the city of Wrocław, *ICCCI 2011, LNAI 6923, Springer* (2011), 191–200.
- [10] R. H. Bordini, J. F. Hubner, M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak using Jason*, John Wiley & Sons, London, UK, 2007.
- [11] M. Behrisch, L. Bieker, J. Erdmann and D. Krajzewicz, SUMO - Simulation of Urban MOBility: An Overview, *SIMUL 2011, The Third International Conference on Advances in System Simulation* (2011), 63–68.

Communication Leading to Coalition Nash Equilibrium I

Takashi MATSUHISA ¹

Ibaraki National College of Technology, Japan

Abstract. This paper is to introduce the new concept of coalition Nash equilibrium of a strategic game, and to show that a communication among the players in a coalition leads to the equilibrium through messages. A coalition Nash equilibrium for a strategic game consists of (1) a subset S of players, (2) independent mixed strategies for each member of S , (3) the conjecture of the actions for the other players not in S with the condition that each member of S maximises his/her expected payoff according to the product of all mixed strategies for S and the other players' conjecture. However, this paper stands on the Bayesian point of view as follows: The players start with the same prior distribution on a state-space. In addition they have private information which is given by a partition of the state space. Each player in a coalition S predicts the other players' actions as the posterior of the others' actions given his/her information. He/she communicates privately their beliefs about the other players' actions through messages among all members in S according to the communication network in S , which message is information about his/her individual conjecture about the others' actions. The recipients update their belief by the messages. Precisely, at every stage each player communicates privately not only his/her belief about the others' actions but also his/her rationality as messages according to a protocol and then the recipient updates their private information and revises her/his prediction. In this circumstance, we show that the conjectures of the players in a coalition S regarding the future beliefs converge in the long run communication, which lead to a coalition Nash equilibrium for the strategic game.

Keywords. Communication, Robust message, Nash equilibrium, Protocol, Conjecture, Non-cooperative game, S5-knowledge model.

1. Introduction

Recently, researchers in economics, AI, and computer science become entertained lively concerns about relationships between knowledge and actions. At what point does an economic agent sufficiently know to stop gathering information and make decisions? There are also concerns about cooperation and knowledge. What is the role of sharing knowledge to making cooperation among agents.

Considering a coalition among agents, we tacitly understand that each agents in the coalition share their individual information and so they commonly know each other. In

¹Corresponding Author: Takashi Matsuhsisa, Department of Natural Science, Ibaraki National College of Technology, Nakane 866, Hitachinaka-shi, Ibaraki 312-8508, Japan. E-mail: mathisa@ge.ibaraki-ct.ac.jp. Partially supported by the Grant-in-Aid for Scientific Research(C)(No.18540153) in the Japan Society for the Promotion of Sciences.

mathematical point of view yet a little is known what structure they have to know commonly. The aim of this paper is to fill the gap. Our point is that in a coalition, the members does not necessary have common-knowledge to each others but they communicate his/her own beliefs on the others to each other through messages.

Many authors have studied the learning processes modeled by Bayesian updating. The papers by E. Kalai and E. Lehrer[1] and J. S. Jordan [2] (and references in therein) indicate increasing interest in the mutual learning processes in games that leads to equilibrium: Each player starts with initial erroneous belief regarding the actions of all the other players. They show the two strategies converges to an ϵ -mixed strategy Nash equilibrium of the repeated game. E. Kalai and E. Lehrer [1] studies two-player repeated games, and they show the two strategies converges to an ϵ -mixed strategy Nash equilibrium of the repeated game if the common prior belief satisfies a certain uniform condition. J. S. Jordan [2] investigates the general convergence result for strategic form games. R. B. Myerson [3] proposes the Bayesian games with mediated communication in which each player is asked to confidentially report his type to the mediator, after getting these reports, the mediator confidentially recommends an action to each player. He characterizes the acceptable correlated equilibria as a subclass of the correlated equilibria in the Bayesian games.

As for as J.F. Nash's fundamental notion of strategic equilibrium is concerned, R.J. Aumann and A. Brandenburger [4] gives epistemic conditions for mixed strategy Nash equilibrium: They show that the common-knowledge of the predictions of the players having the partitional information (that is, equivalently, the **S5**-knowledge model) yields a Nash equilibrium of a game. The paper is in the line of Epistemic Foundation of Interactive Decision Theory, starting by Aumann [5]. In the paper he introduced the formal notion of common-knowledge. However it is not clear just what learning process leads to the equilibrium.

To fill this gap from epistemic point of view, Matsuhisa ([6], [7], [8]) presents his communication system for a strategic game, which leads a mixed Nash equilibrium in several epistemic models. The articles by Matsuhisa ([6], [7]) and the article by Matsuhisa and Strokan [9] together treat the communication system in the **S4**-knowledge model where each player communicates to other players by sending exact information about his/her conjecture on the others' action. In Matsuhisa and Strokan [9], the communication model in the p -belief system is introduced by Monderer and Samet [10]: Each player sends exact information that he/she believes that the others play their actions with probability at least his/her conjecture as messages. Matsuhisa [8] extended the communication model to the case that the sending messages are non-exact information that he/she believes that the others play their actions with probability at least his/her conjecture.

Our research is in the line. The purposes of this paper are to introduce the concept of coalition Nash equilibrium of a strategic game, and to show that a communication among the players in a coalition leads to the equilibrium through messages. A coalition Nash equilibrium for a strategic game consists of (1) a subset S of players, (2) independent mixed strategies for each member of S , (3) the conjecture of the actions for the other players not in S with the condition that each member of S maximizes his/her expected payoff according to the product of all mixed strategies for S and the other players' conjecture.

This paper analyses the solution concept from the Bayesian point of view: The players start with the same prior distribution on a state-space. In addition they have private

information which is given by a partition of the state space. Each player in a coalition S predicts the other players' actions as the posterior of the others' actions given his/her information. He/she communicates privately their beliefs about the other players' actions through messages among all members in S according to the communication network in S , which message is information about his/her individual conjecture about the others' actions. The recipients update their belief by the messages. Precisely, at every stage each player communicates privately not only his/her belief about the others' actions but also his/her rationality as messages according to a protocol and then the recipient updates their private information and revises her/his prediction. In this circumstance, we shall show that

Main theorem. *Suppose that the players in a strategic form game have the knowledge structure associated a partitional information with a common prior distribution. In a communication process of the game according to a protocol with revisions of their beliefs about the other players' actions, the profile of their future predictions converges to a mixed strategy Nash equilibrium of the game in the long run.*

This paper organizes as follows. Section 2 recalls the knowledge structure associated with a partition information structure, and we extend a game on knowledge structure. The communication process for the game is introduced where the players send messages about their conjectures about the other players' action. In Section 3 we give the formal statement of the main theorem (Theorem 1) and will illustrate it by a simple example. In Section 4 we conclude with remarks.

2. The Model

Let Ω be a non-empty *finite* set called a *state-space*, N a set of finitely many *players* $\{1, 2, \dots, n\}$ at least two ($n \geq 2$), and let 2^Ω be the family of all subsets of Ω . Each member of 2^Ω is called an *event* and each element of Ω called a *state*. Let μ be a probability measure on Ω which is common for all players. For simplicity it is assumed that (Ω, μ) is a *finite* probability space with μ full support.²

2.1. Information and Knowledge³

A *partitional information structure* $\langle \Omega, (\Pi_i)_{i \in N} \rangle$ consists of a state space Ω and a class of the mappings Π_i of Ω into 2^Ω such that

- (i) $\{\Pi_i(\omega) | \omega \in \Omega\}$ is a partition of Ω ;
- (ii) $\omega \in \Pi_i(\omega)$ for every $\omega \in \Omega$.

Given our interpretation, an player i for whom $\Pi_i(\omega) \subseteq E$ knows, in the state ω , that some state in the event E has occurred. In this case we say that in the state ω the player i knows E .

²That is; $\mu(\omega) \neq 0$ for every $\omega \in \Omega$.

³C.f.; Bacharach [11], Binmore [12] for the information structure and the knowledge operator.

Definition 1. The knowledge structure $\langle \Omega, (\Pi_i)_{i \in N}, (K_i)_{i \in N} \rangle$ consists of a partitional information structure $\langle \Omega, (\Pi_i)_{i \in N} \rangle$ and a class of i 's knowledge operator K_i on 2^Ω such that $K_i E$ is the set of states of Ω in which i knows that E has occurred; that is,

$$K_i E = \{\omega \in \Omega \mid \Pi_i(\omega) \subseteq E\}.$$

The set $\Pi_i(\omega)$ will be interpreted as the set of all the states of nature that i knows to be possible at ω , and $K_i E$ will be interpreted as the set of states of nature for which i knows E to be possible. We will therefore call Π_i i 's possibility operator on Ω and also will call $\Pi_i(\omega)$ i 's information set at ω .

We record the properties of i 's knowledge operator⁴: For every E, F of 2^Ω ,

- | | | | |
|----------|--|----------|--------------------------------------|
| N | $K_i \Omega = \Omega$ and $K_i \emptyset = \emptyset$; | K | $K_i(E \cap F) = K_i E \cap K_i F$; |
| T | $K_i F \subseteq F$; | 4 | $K_i F \subseteq K_i K_i F$; |
| 5 | $\Omega \setminus K_i(E) \subseteq K_i(\Omega \setminus K_i(E))$. | | |

Remark 1. i 's possibility operator Π_i is uniquely determined by i 's knowledge operator K_i satisfying the above five properties: For $\Pi_i(\omega) = \bigcap_{\omega \in K_i E} E$.

2.2. Game on knowledge structure⁵

By a game G we mean a finite strategic form game $\langle N, (A_i)_{i \in N}, (g_i)_{i \in N} \rangle$ with the following structure and interpretations: N is a finite set of players $\{1, 2, \dots, i, \dots, n\}$ with $n \geq 2$, A_i is a finite set of i 's actions (or i 's pure strategies) and g_i is an i 's payoff function of A into \mathbf{R} , where A denotes the product $A_1 \times A_2 \times \dots \times A_n$, A_{-i} the product $A_1 \times A_2 \times \dots \times A_{i-1} \times A_{i+1} \times \dots \times A_n$. We denote by g the n -tuple (g_1, g_2, \dots, g_n) and by a_{-i} the $(n-1)$ -tuple $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$ for a of A . Furthermore we denote $a_{-I} = (a_i)_{i \in N \setminus I}$ for each $I \subseteq N$. A probability distribution σ_i on A_i is called an i 's mixed strategy for a game G . We denote by $\Delta(A_i)$ the set of all i 's mixed strategies, so we will denote $\Delta(A) = \prod_{i=1}^n \Delta(A_i)$ and $\Delta(A_I) = \prod_{i \in I} \Delta(A_i)$.

Definition 2. A profile $(\sigma_i)_{i \in N}$ of mixed strategies is called a *Nash equilibrium* if for each $i \in N$ and for every $b_i \in A_i$, we have

$$\sum_{a_{-i} \in A_{-i}} g_i(a_i, a_{-i}) \prod_{j \in N \setminus \{i\}} \sigma_i(a_j) \geq \sum_{a_{-i} \in A_{-i}} g_i(b_i, a_{-i}) \prod_{j \in N \setminus \{i\}} \sigma_i(a_j)$$

A probability distribution $\phi_i \in \Delta(A_{-i}) = \Delta(A_{N \setminus \{i\}})$ is said to be i 's overall conjecture (or simply i 's conjecture). For each player j other than i , this induces the marginal distribution on j 's actions; we call it i 's individual conjecture about j (or simply i 's conjecture about j .) Functions on Ω are viewed like random variables in the probability space (Ω, μ) . If \mathbf{x} is a such function and x is a value of it, we denote by $[\mathbf{x} = x]$ (or simply by $[x]$) the set $\{\omega \in \Omega \mid \mathbf{x}(\omega) = x\}$.

The information structure (Π_i) with a common prior μ yields the distribution on $A \subseteq \Omega$ defined by $\mathbf{q}_i(a, \omega) = \mu([a = a] \mid \Pi_i(\omega))$; and the i 's overall conjecture defined by

⁴According to these we can say the structure $\langle \Omega, (K_i)_{i \in N} \rangle$ is a model for the multi-modal logic S5.

⁵C.f., Aumann and Brandenburger [4]

the marginal distribution $\mathbf{q}_i(a_{-i}, \omega) = \mu([a_{-i} = a_{-i}] | \Pi_i(\omega))$ which is viewed as a random variable of φ_i . We denote by $[\mathbf{q}_i = \varphi_i]$ the intersection $\bigcap_{a_{-i} \in A_{-i}} [\mathbf{q}_i(a_{-i}) = \varphi_i(a_{-i})]$ and denote by $[\varphi]$ the intersection $\bigcap_{i \in N} [\mathbf{q}_i = \varphi_i]$. Let \mathbf{g}_i be a random variable of i 's payoff function g_i and \mathbf{a}_i a random variable of an i 's action a_i .

According to the Bayesian decision theoretical point of view, we assume that each player i absolutely knows his/her own actions; i.e., letting $[a_i] := [\mathbf{a}_i = a_i]$, $[a_i] = K_i([a_i])$ (or equivalently, $\Pi_i(\omega) \subseteq [a_i]$ for all $\omega \in [a_i]$ and for every a_i of A_i .) i 's action a_i is said to be *actual* at a state ω if $\omega \in [\mathbf{a}_i = a_i] := \bigcap_{a_i \in A_i} [\mathbf{a}_i = a_i]$; and the profile a_I is said to be actually played at ω if $\omega \in [\mathbf{a}_I = a_I] := \bigcap_{i \in I} [\mathbf{a}_i = a_i]$ for $I \subseteq N$. The pay off functions $\mathbf{g} = (g_1, g_2, \dots, g_n)$ is said to be *actually played* at a state ω if $\omega \in [\mathbf{g} = g] := \bigcap_{i \in N} [\mathbf{g}_i = g_i]$. Let \mathbf{Exp} denote the expectation defined by

$$\mathbf{Exp}(g_i(b_i, \mathbf{a}_{-i}); \omega) := \sum_{a_{-i} \in A_{-i}} g_i(b_i, a_{-i}) \mathbf{q}_i(a_{-i}, \omega).$$

By a *coalition* S we mean S is a non-empty subset of N . Let $(\sigma_i)_{i \in S}$ be a profiles of mixed strategies of G for a coalition S . By S -*expectation* of i 's pay off function g_i at ω we mean

$$\mathbf{Exp}_S(g_i(a_S, \mathbf{a}_{-S}); \omega) := \sum_{a_{-S} \in A_{-S}} g_i(a_S, a_{-S}) (\prod_{i \in S} \sigma_i) \mathbf{q}_i(a_{-S}, \omega).$$

Definition 3. A profile $(\sigma_i)_{i \in S}$ is called a *coalition S-Nash equilibrium* of G if each member i in S maximises his/her $\mathbf{Exp}_S(g_i(a_S, \mathbf{a}_{-S}); \omega)$ for every $\omega \in \Omega$; i.e.: $\mathbf{Exp}_S(g_i(a_S, \mathbf{a}_{-S}); \omega) \geq \mathbf{Exp}_S(g_i(b_S, \mathbf{a}_{-S}); \omega)$ for every b_S in A_S .

A coalition S is said to be *rational* at ω if for every $i \in S$, each i 's actual action a_i maximizes the expectation of his actually played pay off function g_i at ω when the other players actions are distributed according to his conjecture $\mathbf{q}_i(\cdot; \omega)$. Formally, letting $g_i = \mathbf{g}_i(\omega)$ and $a_i = \mathbf{a}_i(\omega)$, $\mathbf{Exp}(g_i(a_i, \mathbf{a}_{-i}); \omega) \geq \mathbf{Exp}(g_i(b_i, \mathbf{a}_{-i}); \omega)$ for every b_i in A_i . Let R_i denote the set of all of the states at which i is rational.

2.3. Protocol⁶

We assume that the players communicate by sending *messages*. Let T be the time horizontal line $\{0, 1, 2, \dots, t, \dots\}$. A *protocol* on a coalition S of a game G is a mapping $\Pr_S : T \rightarrow S \times S, t \mapsto (s(t), r(t))$ such that $s(t) \neq r(t)$. Here t stands for *time* and $s(t)$ and $r(t)$ are, respectively, the *sender* and the *recipient* of the communication which takes place at time t . Simply we call it a *S-protocol*. We consider the protocol as the directed graph whose vertices are the set of all members in S and such that there is an edge (or an arc) from i to j if and only if there are infinitely many t such that $s(t) = i$ and $r(t) = j$.

A protocol \Pr_S is said to be *fair* if the graph is strongly-connected; in words, every player in this protocol communicates directly or indirectly with every other player infinitely often. It is said to contain a *cycle* if there are players i_1, i_2, \dots, i_k with $k \geq 3$ such that for all $m < k$, i_m communicates directly with i_{m+1} , and such that i_k communicates directly with i_1 . The communications is assumed to proceed in *rounds*⁷

⁶C.f.: Parikh and Krasucki [13]

⁷There exists a time m such that for all t , $\Pr_S(t) = \Pr_S(t+m)$. The *period* of the protocol is the minimal number of all m such that for every t , $\Pr_S(t+m) = \Pr_S(t)$.

2.4. Communication on coalition

Let S be a coalition of G . A *coalition S-communication process* $\pi_S(G)$ with revisions of players' conjectures $(\varphi_i^t)_{(i,t) \in S-T}$ according to a protocol for a game G is a tuple

$$\pi_S(G) = \langle G, (\Omega, \mu) \text{Pr}_S, (\Pi_i^t)_{i \in S}, (K_i^t)_{i \in S}, (\varphi_i^t)_{(i,t) \in S-T} \rangle$$

with the following structures: the players have a common prior μ on Ω , the protocol Pr_S among N , $\text{Pr}_S(t) = (s(t), r(t))$, is fair and it satisfies the conditions that $r(t) = s(t+1)$ for every t and that the communications proceed in rounds. The revised information structure Π_i^t at time t is the mapping of Ω into 2^Ω for player $i \in S$. If $i = s(t)$ is a sender at t , the message sent by i to $j = r(t)$ is M_j^t . An n -tuple $(\varphi_i^t)_{i \in S}$ is a revision process of individual conjectures. These structures are inductively defined as follows:

- Set $\Pi_i^0(\omega) = \Pi_i(\omega)$.
- Assume that Π_i^t is defined. It yields the distribution $\mathbf{q}_i^t(a, \omega) = \mu([a = a] | \Pi_i^t(\omega))$. Whence

R_i^t denotes the set of all the state ω at which i is *rational* according to his conjecture $\mathbf{q}_i^t(\cdot; \omega)$; that is, each i 's actual action a_i maximizes the expectation of his pay off function g_i being actually played at ω when the other players actions are distributed according to his conjecture $\mathbf{q}_i^t(\cdot; \omega)$ at time t .⁸

The message $M_j^t : \Omega \rightarrow 2^\Omega$ sent by the sender i at time t is defined by

$$M_j^t(\omega) = \bigcap_{a_{-i} \in A_{-i}} \{ \xi \in \Omega \mid \mathbf{q}_i^t(a_{-i}, \xi) = \mathbf{q}_i^t(a_{-i}, \omega) \}.$$

Then:

- The revised knowledge operator $K_i^t : 2^\Omega \rightarrow 2^\Omega$ is defined by

$$K_i^t(E) = \{ \omega \in \Omega \mid \Pi_i^t(\omega) \subseteq E \}.$$

- The revised partition Π_i^{t+1} at time $t+1$ is defined as follows:

$$\begin{aligned} \Pi_i^{t+1}(\omega) &= \Pi_i^t(\omega) \cap M_{s(t)}^t(\omega) \text{ if } i = r(t); \\ \Pi_i^{t+1}(\omega) &= \Pi_i^t(\omega) \text{ otherwise,} \end{aligned}$$

- The revision process $(\varphi_i^t)_{(i,t) \in S-T}$ of conjectures is inductively defined by the following way:

Let $\omega_0 \in \Omega$, and set $\varphi_{s(0)}^0(a_{-s(0)}) := \mathbf{q}_{s(0)}^0(a_{-s(0)}, \omega_0)$

Take $\omega_1 \in M_{s(0)}^0(\omega_0) \cap K_{r(0)}([g_{s(0)}] \cap R_{s(0)}^0)$,⁹ and set $\varphi_{s(1)}^1(a_{-s(1)}) := \mathbf{q}_{s(1)}^1(a_{-s(1)}, \omega_1)$

Take $\omega_{t+1} \in M_{s(t)}^t(\omega_t) \cap K_{r(t)}([g_{s(t)}] \cap R_{s(t)}^t)$, and set $\varphi_{s(t+1)}^{t+1}(a_{-s(t+1)}) := \mathbf{q}_{s(t+1)}^{t+1}(a_{-s(t+1)}, \omega_{t+1})$.

⁸Formally, letting $g_i = g_i(\omega)$, $a_i = \mathbf{a}_i(\omega)$, the expectation at time t , Exp^t , is defined by $\text{Exp}^t(g_i(a_i, \mathbf{a}_{-i}); \omega) := \sum_{a_{-i} \in A_{-i}} g_i(a_i, a_{-i}) \mathbf{q}_i^t(a_{-i}, \omega)$. A player $i \in S$ is said to be S -rational according to his conjecture $\mathbf{q}_i^t(\cdot, \omega)$ at ω if for all $b_i \in A_i$, $\text{Exp}^t(g_i(a_i, \mathbf{a}_{-i}); \omega) \geq \text{Exp}^t(g_i(b_i, \mathbf{a}_{-i}); \omega)$.

⁹We denote $[g_i] := [\mathbf{g}_i = g_i]$

The specification is that a sender $s(t)$ at time t informs the recipient $r(t)$ his/her prediction about the other players' actions as approximate information of his/her individual conjecture to an accuracy ε . The recipient revises her/his information structure under the information. She/he predicts the other players action at the state where the player knows that the sender $s(t)$ is rational, and she/he informs her/his the predictions to the other player $r(t+1)$.

We denote by ∞ a sufficient large $\tau \in T$ such that for all $\omega \in \Omega$, $\mathbf{q}_i^\tau(\cdot; \omega) = \mathbf{q}_i^{\tau+1}(\cdot; \omega) = \mathbf{q}_i^{\tau+2}(\cdot; \omega) = \dots$. Hence we can write \mathbf{q}_i^τ by \mathbf{q}_i^∞ and φ_i^τ by φ_i^∞ .

Remark 2. This communication model is a variation of the model introduced by Matsuhisa [6].

3. The Result

Before proceeding with stating our result, we will give the illustrative example: This shows that the notion of common-knowledge cannot control to form a coalition Nash equilibrium, even though it can do for Nash equilibrium for a game (Aumann and Brandenburger [4]).

Example 1. Let us consider the three persons game $G = \langle N, (A_i)_{i \in N}, (g_i)_{i \in N} \rangle$ as follows;

- The set of players $N = \{1, 2, 3\}$:
- The action sets $A_1 = \{H, T\}, A_2 = \{H, T\}, A_3 = \{W, E\}$:

The pay-off functions g_1, g_2, g_3 are given by Table 1: The game G has the unique Nash

Table 1.

W	h	t	E	h	t
H	1, 0, 2	0, 1, 2	H	1, 0, 3	0, 1, 0
T	0, 1, 2	1, 0, 2	T	0, 1, 0	1, 0, 3

equilibrium $(\frac{1}{2}H + \frac{1}{2}T, \frac{1}{2}h + \frac{1}{2}t, W)$. \square

We let start the situation: *Each player knows his/her own actions, but he/she cannot know the other players' action*. To model the situation we introduce the game G as a Bayesian game equipped with the below information partition $(\Pi_i)_{i=1,2,3}$:

- The state space $\Omega = \{\omega_1, \omega_2, \dots, \omega_8\}$:
- μ is the equal probability measure on 2^Ω ; i.e., $\mu(\omega) = \frac{1}{8}$:
- The partitions $(\Pi_i)_{i=1,2,3}$ on Ω :

The partition Π_1 on Ω :

$$\Pi_1(\omega) = \{\omega_i \mid i = 1, 2, 5, 6\} \text{ for } \omega = \omega_i (i = 1, 2, 5, 6)$$

$$\Pi_1(\omega) = \{\omega_i \mid i = 3, 4, 7, 8\} \text{ for } \omega = \omega_i (i = 3, 4, 7, 8)$$

The partition Π_2 on Ω :

$$\Pi_2(\omega) = \{\omega_i \mid i = 1, 3, 5, 7\} \text{ for } \omega = \omega_i(i = 1, 3, 5, 7)$$

$$\Pi_2(\omega) = \{\omega_i \mid i = 2, 4, 6, 8\} \text{ for } \omega = \omega_i(i = 2, 4, 6, 8)$$

The partition Π_3 on Ω :

$$\Pi_3(\omega) = \{\omega_1, \omega_2, \omega_3, \omega_4\} \text{ for } \omega = \omega_i(i = 1, 2, 3, 4)$$

$$\Pi_3(\omega) = \{\omega_5, \omega_6, \omega_7, \omega_8\} \text{ for } \omega = \omega_i(i = 5, 6, 7, 8)$$

- \mathbf{a}_i is defined by

$$\mathbf{a}_1(\omega) = H \quad \text{for } \omega = \omega_i(i = 1, 2, 5, 6)$$

$$\mathbf{a}_1(\omega) = T \quad \text{for } \omega = \omega_i(i = 3, 4, 7, 8)$$

$$\mathbf{a}_2(\omega) = h \quad \text{for } \omega = \omega_i(i = 1, 3, 5, 7)$$

$$\mathbf{a}_2(\omega) = t \quad \text{for } \omega = \omega_i(i = 2, 4, 6, 8)$$

$$\mathbf{a}_3(\omega) = W \quad \text{for } \omega = \omega_i(i = 1, 2, 3, 4)$$

$$\mathbf{a}_3(\omega) = E \quad \text{for } \omega = \omega_i(i = 5, 6, 7, 8)$$

We can observe that the conjectures $\varphi_i(a_j) = \mathbf{q}_i(a_j; \omega_5)$ at ω_5 are

- $\varphi_2(a_1) = \varphi_3(a_1) = \frac{1}{2}H + \frac{1}{2}T$
- $\varphi_1(a_2) = \varphi_3(a_2) = \frac{1}{2}h + \frac{1}{2}t$
- $\varphi_1(a_3) = \varphi_2(a_3) = \frac{1}{2}W + \frac{1}{2}E.$

This shows that for each player i , any the other players than i must agree on every i ' actions, but these distributions $(\varphi_3(a_1), \varphi_1(a_2), \varphi_2(a_3))$ cannot form the Nash equilibrium for G , but $(\varphi_2(a_1), \varphi_1(a_2)) = (\varphi_3(a_1), \varphi_1(a_2))$ forms a coalition $\{1, 2\}$ -Nash equilibrium. However $(\varphi_3(a_1), \varphi_1(a_3))$ is not a $\{1, 3\}$ -Nash equilibrium. It should be noted that $(\varphi_3(a_1), \varphi_1(a_3))$ is commonly known among $\{1, 3\}$, and so the notion of common-knowledge cannot always imply a coalition Nash equilibrium.

Let S be the coalition $\{1, 3\}$, and consider the coalition S -communication process for game G equipped with the protocol $\Pr : T \rightarrow S = \{1, 3\}$. After 2 rounds communication, the below information partition can be obtained: $(\Pi_i)_{i=1,2,3}$:

- The partitions $(\Pi_i^\infty)_{i \in N}$ on Ω :

The partition Π_1 on Ω :

$$\Pi_1^\infty(\omega) = \{\omega_i \mid i = 1, 2\} \text{ for } \omega = \omega_i(i = 1, 2)$$

$$\Pi_1^\infty(\omega) = \{\omega_i \mid i = 5, 6\} \text{ for } \omega = \omega_i(i = 5, 6)$$

$$\Pi_1^\infty(\omega) = \{\omega_i \mid i = 3, 4\} \text{ for } \omega = \omega_i(i = 3, 4)$$

$$\Pi_1^\infty(\omega) = \{\omega_i \mid i = 7, 8\} \text{ for } \omega = \omega_i(i = 7, 8)$$

The partition Π_2^∞ on Ω is the same as the initial partition Π_2 :

The partition Π_3^∞ on Ω is the same as the initial partition Π_3 :

We can observe that the conjectures $\varphi_i^\infty(a_j) = \mathbf{q}_i^\infty(a_j; \omega)$ at ω_5 are

- $\varphi_2^\infty(a_1) = \varphi_3^\infty(a_1) = \frac{1}{2}H + \frac{1}{2}T$
- $\varphi_1^\infty(a_2) = \varphi_3^\infty(a_2) = \frac{1}{2}h + \frac{1}{2}t$
- $\varphi_1^\infty(a_3) = \varphi_2^\infty(a_3) = W$,

and these distributions $(\varphi_3^\infty(a_1), \varphi_1^\infty(a_3))$ is a $\{1, 3\}$ -Nash equilibrium. Furthermore, $(\varphi_2^\infty(a_1), \varphi_1^\infty(a_2), \varphi_1^\infty(a_3))$ forms the Nash equilibrium for G . \square

We can now state the main theorem, and we omit the proof because of the limitation of pages. It will be appeared in a future paper (Matsuhisa [14]).

Theorem 1. Suppose that the players in a strategic form game G have the knowledge structure with μ a common prior. Let S be a coalition in a game G . In the coalition S -communication process $\pi_S(G)$ according to a protocol Pr_S among all members in S , the $|S|$ -tuple of their conjectures $(\varphi_i^t)_{(i,t) \in S \times T}$ converges to a coalition S -Nash equilibrium of the game in finitely many rounds. \square

Remark 3. When S is the ground coalition N , the above theorem shows that the conjectures of the players leads to a Nash equilibrium through communication. This is shown in Matsuhisa [6].

4. Concluding remarks

Conclusion

This paper introduces the solution concept ‘coalition Nash equilibrium’, and this solution can be formed in sharing information by communication not by common-knowledge. The illustrated example shows that common-knowledge cannot play such role as for mixed strategy Nash equilibrium; the profile of conjectures of a coalition may not yield a coalition Nash equilibrium even when the profile is commonly known among the all members of the coalition. In this paper we adopt the knowledge revisions model by Parikh and Krasucki [13] as a communication model, and the main theorem shows that communication instead common-knowledge plays an essential role to form a coalition Nash equilibrium. In fact, we have observed that in a communication process with revisions of players’ beliefs about the other actions among all the members in a coalition, their predictions induces a coalition Nash equilibrium of the game in the long run.

Further research

It well ends this paper some appraisals on further research. For making a mixed strategy Nash equilibrium by communication concerned, Matsuhisa [6] and [7] established the same assertion in the **S4**-knowledge model. Furthermore Matsuhisa [15] showed a similar result for ϵ -mixed strategy Nash equilibrium of a strategic form game in the **S4**-knowledge model, which gives an epistemic aspect in Theorem of E. Kalai and E. Lehrer [1]. We can easily extend Theorem 1 to these cases.

In the communication model presented we assume rationality of each player; i.e., when each maximises his/her own expectation at every state where he/she will send message. We have not touched on the big problem: How to control each player's rationality by sharing knowledge. this is our next agendum.

References

- [1] Kalai, E., and Lehrer, E.: Rational learning to mixed strategy Nash equilibrium, *Econometrica* **61** (1993) 1019–1045.
- [2] Jordan, J. S.: Bayesian learning in normal form games, *Games and Economic Behavior* **3** (1991) 60–81.
- [3] Myerson, R. B.: Acceptable and predominant correlated equilibria *International Journal of Game Theory*, **15** (1986) 133–154.
- [4] Aumann, R. J., and Brandenburger, A.: Epistemic conditions for mixed strategy Nash equilibrium, *Econometrica* **63** (1995) 1161–1180.
- [5] Aumann, R.: Agreeing to disagree. *Annals of Statistics* **4** (1976) pp.1236-1239.
- [6] Matsuhisa, T.: Communication leading to mixed strategy Nash equilibrium I, T. Maruyama (eds) *Mathematical Economics*, Suri-Kaiseki-Kenkyusyo Kokyuroku **1165** (2000) 245–256.
- [7] Matsuhisa, T.: Communication leading to a Nash equilibrium without acyclic condition (S4-knowledge case), M. Bubak et al (eds) *International Conference on Computer Science*, Springer Lecture Notes in Computer Science **3039** (2004) 884–891.
- [8] Matsuhisa, T.: Bayesian communication under rough sets information, C. J. Butz et al (eds) *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, WI-ITA 2006 Workshop Proceedings, IEEE Computer Society (2006) 378–381.
- [9] Matsuhisa, T., and Strokan, P.: Bayesian belief communication leading to a Nash equilibrium in belief, Deng, X. and Ye, Y. (eds) *Internet and Network Economics*, Springer Lecture Notes in Computer Science **3828** (2005) 299–306.
- [10] Monderer, D., and Samet, D.: Approximating common knowledge with common beliefs, *Games and Economic Behaviors* **1** (1989) 170–190.
- [11] Bacharach, M.: Some extensions of a claim of Aumann in an axiomatic model of knowledge, *Journal of Economic Theory* **37** (1985) 167–190.
- [12] Binmore, K.: *Fun and Games*. xxx+642pp. Lexington, Massachusetts USA: D. C. Heath and Company (1992).
- [13] Parikh R., and Krasucki, P.: Communication, consensus, and knowledge, *Journal of Economic Theory* **52** (1990) 178–189.
- [14] Matsuhisa, T.: Communication leading to coalition Nash equilibrium II – S5n knowledge case –, Preprint, Working Paper, Ibaraki National College of Technology (2013) 13pp.
- [15] Matsuhisa, T.: Communication leading to epsilon-mixed strategy Nash equilibrium, Working paper (2001). The extended abstract was presented in the XIV Italian Meeting of Game Theory and Applications (IMGTA XIV), July 11-14, 2001.

Agent-Based Optimization

An Agent-Based Cooperative Population Learning Algorithm for Vehicle Routing Problem with Time Windows

Dariusz BARBUCHA¹

Department of Information Systems, Gdynia Maritime University, Poland

Abstract. Population-based metaheuristics, mostly inspired by biological or social phenomena, belong to a widely used class of approaches suitable for solving complex hard optimization problems. Their effectiveness has been confirmed for many real-time instances of different optimization problems. This paper proposes an Agent-Based Cooperative Population Learning Algorithm for the Vehicle Routing Problem with Time Windows, where the search for solutions is divided into stages, and different learning/improvement procedures are used at each stage. These procedures are based on a set of heuristics (represented as software agents) which are run under the cooperation schema defined separately for each stage. Computational experiment, which has been carried out, confirmed the effectiveness of the proposed approach.

Keywords. agent-based optimization, population-based methods, collective problem solving, vehicle routing problems

1. Introduction

Last years a spectrum of methods used to solve NP-hard optimization problems has evolved from relatively simple constructive methods to more complex local search methods. The first group, including fast and simple methods, offers competitive behavior and results only for a relatively small instances. On the other hand, using local search methods for solving instances of a given problem gives an opportunity for improvement of results. Although a significant improvement of results produced by these methods can be often observed as compared with the constructive methods, the results are still not satisfactory, especially for more complex or practical instances of many optimization problems. The main disadvantage of using the local search methods is their tendency to get stuck in local optima, often located far from the globally best solution.

In order to avoid this inconvenience of the local search methods, a group of metaheuristic approaches have been proposed for solving difficult optimization problems. Typical classification of these methods distinguishes: single solution based metaheuristics and population-based ones [1]. The first group of methods, including, for example, Tabu Search [2], Simulated Annealing [3] or Greedy Randomized Adaptive Search [4]

¹Corresponding Author: Dariusz Barbucha, Department of Information Systems, Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland; E-mail: d.barbucha@wpit.am.gdynia.pl

procedures, concentrates on improving a single solution (individual). On the other hand, population-based metaheuristics handle a population of individuals that evolves with the help of information exchange procedures. A class of the population based metaheuristics, mostly inspired by biological or social processes, includes Evolutionary Algorithms [5], Differential Evolution [6] methods, Gene Expression Programming approaches [7], Scatter Search methods [8], Ant Colony Optimization algorithms [9], and Particle Swarm Optimization algorithms [10].

An interesting approach which extends the population-based metaheuristics group has been proposed by Jędrzejowicz [11] under the name Social Learning Algorithm, next renamed to Population Learning Algorithm (PLA). In contrast to the most population-based methods, PLA has been inspired by an analogy to the social education systems, where a massive number of individuals enter the system, the learning process is divided into stages, more advanced stages are entered by a decreasing number of individuals from the initial population and the final stages can be reached by only a few of the best. It was shown that PLA could be useful to solve some difficult optimization problems like the generalized segregated storage problem [12], quadratic assignment problem [13] and some difficult scheduling problems [11,14].

In its classical version, PLA divides process of solving the problem into stages, in which a given optimization problem is solved using a set of independent learning/improvement procedures, each procedure at a single stage. As a rule, the procedures used at higher stages are more complex than ones, used in earlier stages. This paper focuses on incorporating a *cooperative problem solving* paradigm into the original PLA, which, in fact, reflects additional important features of educational systems, like team working and collective problem solving. According to Blum and Roli [15], the cooperation is meant here as a problem-solving strategy, consisting of a search performed by a team of agents that exchange information about states, models, entire sub-problems, solutions or other search space characteristics. In particular, the proposed Cooperative Population Learning Algorithm (CPLA) extends the classic PLA allowing the possibility of using a few procedures at each stage of learning/improvement (it reflects different education techniques, methods used in contemporary education systems), which can cooperate during the process of search. The cooperation is organized in indirect form, which means that results obtained by one heuristic (or team of heuristics) can be shared with other heuristics (or teams of heuristics), engaged in process of solving instances of given problem. The architecture of the proposed CPLA at a single stage is based on the asynchronous team (A-Team) [16] implementation proposed in [17] and extended in [18]. In general, A-Team is a multi agent architecture, where a collection of software agents solve a problem by dynamically evolving a population of solutions. Each agent works to create, modify or remove solutions from the population, and the quality of the solutions gradually evolves over time as improved solutions are added and poor solutions are removed. The paper includes a study and experimental validation of the proposed CPLA designed to solve instances of the Vehicle Routing Problem with Time Windows.

The rest of the paper is organized as follows. Section 2 outlines the main features of CPLA. In Sections 3 and 4 a formulation of VRPTW and an implementation of CPLA for VRPTW details are included, respectively. Results of the computational experiment which has been carried out are presented in Section 5. Section 6 concludes the paper.

2. Concept of the Cooperative Population Learning Algorithm

As other population-based methods, the proposed Cooperative Population Learning Algorithm handles a population of individuals, representing coded solutions of the considered problem. The process of solving an instance of the problem is divided into learning/improvement stages, where a set of heuristics working under the predefined cooperation scheme is used at each stage. Initially, a population including a large number of individuals is created, either randomly or using a simple constructive heuristic, and next it is stored in sharable memory. Once the initial population has been generated its individuals enter the first learning stage. It involves applying a set of possibly basic and elementary heuristics to each individual. The process of improvement individuals is organized as a sequence of steps in which a selected individual is taken from the population (memory), next, it is improved by the selected heuristic, and after an attempted improvement it is stored back in the sharable memory containing population of solutions. The above process of learning/improvement is repeated at this stage until a predefined stopping criterion is met. Next, the improved individuals are evaluated and better ones pass to the next stage and the remaining are dropped from the process. At the following stages the whole cycle is repeated. Individuals are subject to improvement and learning and the selected ones are again promoted to the higher stage. At the final stage the best and the brightest are evaluated in order to select a solution to the problem at hand.

It is easy to see, that the number of individuals stored in the common (sharable) memory successively decreases in subsequent stages. Also, the population successively evolves from the initially generated pool of solutions through intermediate trial solutions obtained during the search process in the first stage up to the stages when only a higher quality solutions are involved in the search process. Average quality of individuals stored in population is gradually improved and the best solution stored in the memory is taken as the final solution of the given problem instance after the final stage of computation.

The pseudocode of CPLA in presented as Algorithm 1. In general, this CPLA scheme can be used for solving different optimization problems, but as in case of other population-based approaches, in order to use it for solving a specific problem, several elements need to be defined. Hence, any cooperative population learning algorithm applied to a particular optimization problem must include such elements as definition of individuals, a method of evaluating an individual (fitness), size of the initial population (*initPopSize*) and the method of creating individuals belonging to it. Moreover, for each stage k , where $k = 1, 2, \dots, noSt$ ($noSt$ - number of stages), a *learning/improvement procedure* $L(k)$ and a *promotion criterion* $PC(k)$ should be defined. Each $L(k)$ includes a set of heuristics - $H(k)$, dedicated for solving instances of the problem at stage k , and a cooperation scheme - $CS(k)$, under which heuristics belonging to $H(k)$ are working ($L(k) = \langle H(k), CS(k) \rangle$). Each $PC(k)$ defines the rule for promotion most promising individuals to the next stage.

This paper focuses on using CPLA for solving a Vehicle Routing Problem with Time Windows.

3. Vehicle Routing Problem with Time Windows

The Vehicle Routing Problem with Time Windows (VRPTW) can be formulated as the problem of determining optimal routes through a given set of locations (customers) and

Algorithm 1 General scheme of Cooperative Population Learning Algorithm (CPLA)**Require:**

- $initPopSize$ - initial population size,
- $noSt$ - number of learning stages,
- $L = \{L(1), L(2), \dots, L(noSt)\}$ - set of learning/improvement procedures used at each stage, where each $L(k)$ includes: $H(k)$ - a set of heuristics (agents) and $CS(k)$ - a cooperation scheme used at stage k ($k = 1, 2, \dots, noSt$)
- $PC = \{PC(1), PC(2), \dots, PC(noSt)\}$ - set the promotion criteria after each stage

Ensure:

- s_best - best solution found
- 1: Generate the initial population of individuals $P = \{p_1, p_2, \dots, p_{initPopSize}\}$ and store it in the common memory
 - 2: **for** $k = 1 \dots noSt$ **do**
 - 3: Apply the learning/improvement procedure $L(k)$ using the set of heuristics $H(k)$ under the cooperation scheme $CS(k)$
 - 4: **for** each individual $p_j \in P$ **do**
 - 5: **if** p_j do not pass the promotion criterion $PC(k)$ **then**
 - 6: $P \leftarrow P \setminus \{p_j\}$
 - 7: **end if**
 - 8: **end for**
 - 9: **end for**
 - 10: **return** the overall best solution s_best

defined on an undirected graph $G = (V, E)$, where $V = \{0, 1, \dots, N\}$ is the set of nodes and $E = \{(i, j) | i, j \in V\}$ is a set of edges. Node 0 is a central depot with NV identical vehicles of capacity W . Each other node $i \in V \setminus \{0\}$ denotes customer characterized by coordinates in Euclidean space (x_i, y_i) and a non-negative demand d_i . Each link $(i, j) \in E$ denotes the shortest path from customer i to j and is described by the cost c_{ij} of travel from i to j by shortest path ($i, j \in V$). It is assumed that $c_{ij} = c_{ji}$ ($i, j \in V$). It is also often assumed that c_{ij} is equal to travel time t_{ij} .

Moreover, in VRPTW, with each customer $i \in V$, a time window $[e_i, l_i]$ wherein the customer has to be supplied, is associated. Here e_i is the earliest possible departure (ready time), and l_i - the latest time the customer has to be started to serve. A time window at the depot $([e_0, l_0])$ is called the scheduling horizon. Let s_i be the service time at the customer $i \in V \setminus \{0\}$.

The goal is to minimize the vehicle fleet and the sum of travel time and waiting time needed to supply all customers in their required time windows (minimization of the fleet size is considered to be the primary objective of the VRPTW), satisfying the following constraints:

- each route starts and ends at the depot,
- each customer $i \in V \setminus \{0\}$ is serviced exactly once by a single vehicle,
- the total load on any vehicle associated with a given route does not exceed vehicle capacity,

- each customer $i \in V$ has to be supplied within a time window $[e_i, l_i]$ associated to it (a vehicle arriving before the lower limit of the time window causes additional waiting time on the route),
- each route must start and end within the time window associated with the depot.

A review of different methods proposed for solving VRPTW can be found in two papers of Braysy and Gendreau [19] and [20].

4. Cooperative Population Learning Algorithm for the VRPTW

Elements of the CPLA dedicated for solving VRPTW have been set as follows.

4.1. Individual

Each solution is *coded* using permutation of N numbers (representing customers) with additional 0 delimiters. A part of individual between these delimiters reflects the order in which customers are visited by one vehicle within the selected route.

4.2. Creating the Initial Population

Method of creating a single individual to be incorporated into the initial population is based on the Solomon's *II* constructive heuristic for VRPTW [21]. But, opposite to the Solomon's approach, where two initialization criteria were tested (the farthest unrouted customer and the uncounted customer with the earliest deadline), here, creation of each route starts from a randomly selected unrouted customer. The process of creating the whole initial population is repeated until *initPopSize* individuals have been generated. In the presented implementation, the size of the initial population (*initPopSize*) has been set to 200.

4.3. Fitness

Each individual from the population is evaluated using the *fitness* function, which value is calculated as the sum of the *number of vehicles/costs* related to each permutation part (vehicles route).

4.4. Learning/improvement Procedures and Promotion Criteria

The *number of stages* of learning has been set to 3. As it was said, at each stage the presented procedures are based on A-Team [16] implementation proposed in [17,18]. Main parts of CPLA for VRPTW with architectures used in each stage are presented in Figure 1.

The following *learning/improvement procedures* (each including: *a set of heuristics* and *cooperation schemes*), as well as *promotion strategies* have been used at each stage.

Stage 1

A large number (*initPopSize*) of individuals enter the first stage.

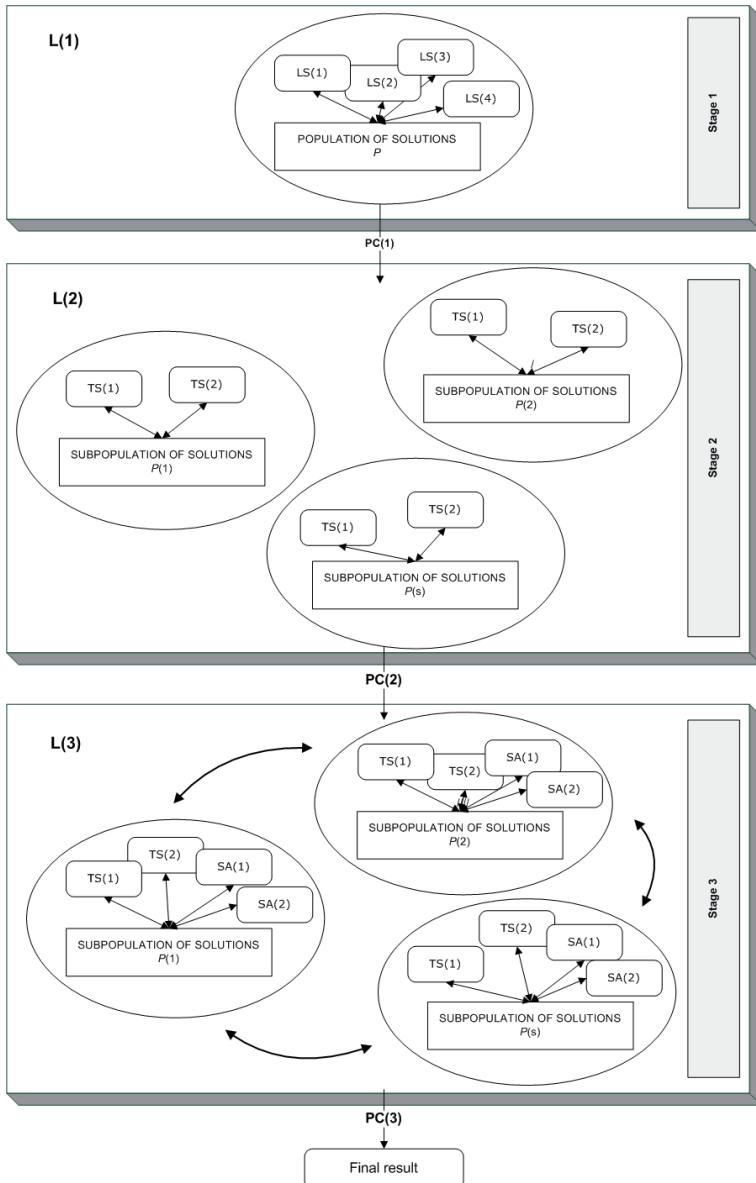


Figure 1. An architecture of the proposed CPLA approach for VRPTW

Learning/improvement procedure L(1): The process of learning/improvement is performed by the set of four Local Search heuristics (represented as software agents) ($H(1)$), based on the following moves:

- a single customer from randomly selected position of individual is moved to another, randomly selected position ($LS(1)$)
- two customers from randomly selected positions of individual are exchanged ($LS(2)$)

- two randomly selected routes are disconnected and the remaining parts from different routes are reconnected again ($LS(3)$)
- two edges from two randomly selected routes are exchanged ($LS(4)$)

The whole process is performed several times in a cycle and includes the steps of *selecting* randomly an individual from the population, *improving* it by a selected heuristic, and *accepting* it (if it has been improved) and merging with the population replacing the first found worse individual from the current population. If a worse individual can not be found within 5 reviews (where review is understood as a search for the worse individual after an improved solution is returned) then the worst individual in the common memory is replaced by the randomly generated one, representing a feasible solution. The process stops after reaching a predefined stopping criterion (3 minutes of computation in this case).

In this stage, cooperation inside the population of individuals ($CS(1)$) has a vertical (indirect) form. It means that agents representing heuristics operate on solutions stored in the common, sharable memory and they work on results obtained by another agents, but information is not exchanged directly between agents representing heuristics but through the common memory containing the population of individuals.

Promotion criterion PC(1): Only individuals with fitness above the average fitness of the whole population enter the next stage.

Stage 2

At this stage, the population is divided into s subpopulations (teams), and each subpopulation is taught using the same learning/improvement procedure $L(2)$.

Learning/improvement procedure L(2): The procedure includes the same set of steps 1-3 as defined at stage 1, but the set of heuristics (represented by agents) consists here of two Tabu Search implementations ($TS(1)$, $TS(2)$) with two moves defined in the same way as in the $LS(1)$ and $LS(2)$ local search methods defined at stage 1. Cooperation inside each subpopulation of individuals has the same form as the previous one. Additionally, each team produces results independently and during the search it does not communicate with other teams.

Promotion criterion PC(2): After preliminary experiments, it has been decided that only 1/3 best individuals from each subpopulation are promoted to the next stage.

Stage 3

And finally, at this stage, individuals forming the same subpopulations as in stage 2 are taught using the following learning/improvement procedure $L(3)$.

Learning/improvement procedure L(3): The set of two Tabu Search and two Simulated Annealing procedures ($TS(1)$, $TS(2)$, $SA(1)$, $SA(2)$) are used by each team. These procedures include two moves defined in the same way as in the $LS(1)$ and $LS(2)$ local search methods defined at stage 1. The form of cooperation inside each subpopulation stays the same as in previous stages, but opposite to stage 2, now a periodical exchange of some solutions between subpopulations is allowed. In the presented implementation it has been assumed that every 0.3 minutes the best solution from each source population is sent to adjacent population (one-way ring topology) and replaces the worst solution in

the target population. Communication implemented at stage 3 has vertical communication form inside each population and horizontal communication between subpopulations of individuals.

Promotion criterion PC(3): The overall best result from all subpopulations is taken as the final solution found for the task.

5. Computational Experiment

The computational experiment has been carried out in order to evaluate to what extent the proposed approach produces results of a good quality, measured as the relative error - RE (in %) from the best known solutions identified by heuristics algorithms, reported in [24].

The approach has been implemented in Java programming language and using JADE (Java Agent Development Framework) [22] - a software framework proposed by TILAB [23] for the development and run-time execution of peer-to-peer applications.

The experiment involved six datasets of instances of Solomon (R1, R2, C1, C2, RC1, RC2) [21] including 100 customers and available at [24]. Each instance was repeatedly solved 10 times. All computations have been carried out on PC computer Intel Core i5-2540M CPU 2.60 GHz with 8 GB RAM running under MS Windows 7 operating system.

Results of the experiment are presented in Table 1. For each instance, the table includes its name, best solution identified till now by heuristics proposed by other authors for solving VRPTW (available in [24]) and best solution obtained by the proposed CPLA approach. Each solution of VRPTW is represented by the number of vehicles and the distance covered by vehicles serving customers.

Analysis of the results presented in Table 1 allows one to make several interesting observations. The first observation is that the results produced by the proposed CPLA are competitive with the best known approximate solutions identified by other heuristics. Taking into account the total distance of all vehicles serving customers, more than half of the results obtained by CPLA is equal to or better than the best known approximate solutions in each group of instances (C, R, RC): 9 out of 17 in group C, 13 out of 23 in group R, and 9 out of 16 in group RC (all these results are emphasized using bold font in Table 1). Although in group C all such results are equal to the best known approximate solutions, in rest groups they are better. On the other hand, taking into account the second objective (total number of vehicle used), it is worth mentioning that in case of group C, all results obtained by CPLA with distance equal to the best known approximate results have also the same number of vehicles used. Unfortunately, in case of groups R and RC, a slight increase of the number of vehicles (1-2 vehicles) is observed.

By focusing observation on CPLA results which are worse than the best known approximate results, the differences between results identified by CPLA and the best ones do not exceed 1% for most instances from group C, and 2% for most instances from groups R and RC. An error greater than 2% has been observed only for instances C104, R101, R102, R106, R112, RC103, RC104 and RC204.

Table 1. Results obtained by the proposed CPLA (min. distance and the number of vehicles serving customers)

Instance	VH[best/CPLA]	Dist[best/CPLA]	Instance	VH[best/CPLA]	Dist[best/CPLA]
C101	10/10	828.94/828.94	C201	3/3	591.56/591.56
C102	10/10	828.94/828.94	C202	3/3	591.56/591.56
C103	10/10	828.06/833.08	C203	3/3	591.17/591.56
C104	10/10	824.78/853.35	C204	3/3	590.60/600.82
C105	10/10	828.94/828.94	C205	3/3	588.88/588.88
C106	10/10	828.94/828.94	C206	3/3	588.49/589.51
C107	10/10	828.94/828.94	C207	3/3	588.29/589.30
C108	10/10	828.94/828.94	C208	3/3	588.32/597.57
C109	10/10	828.94/833.39			
R101	19/20	1645.79/1684.54	R201	4/5	1252.37/1211.48
R102	17/17	1486.12/1525.86	R202	3/4	1191.70/1055.39
R103	13/14	1292.68/1244.30	R203	3/4	939.54/920.54
R104	9/11	1007.24/1015.48	R204	2/3	825.52/770.32
R105	14/15	1377.11/1395.96	R205	3/4	994.42/986.48
R106	12/14	1251.98/1291.06	R206	3/3	906.14/921.81
R107	10/11	1104.66/1098.90	R207	2/3	893.33/852.23
R108	9/10	960.88/964.64	R208	2/3	726.75/729.41
R109	11/12	1194.73/1184.24	R209	3/4	909.16/900.59
R110	10/12	1118.59/1112.36	R210	3/4	939.34/944.63
R111	10/12	1096.72/1086.93	R211	2/3	892.71/796.24
R112	9/10	982.14/1002.80			
RC101	14/16	1696.94/1666.56	RC201	4/5	1406.91/1361.04
RC102	12/14	1554.75/1505.95	RC202	3/5	1367.09/1162.68
RC103	11/12	1261.67/1329.73	RC203	3/4	1049.62/976.43
RC104	10/11	1135.48/1175.57	RC204	3/3	798.41/815.16
RC105	13/15	1629.44/1590.44	RC205	4/5	1297.19/1270.13
RC106	11/13	1424.73/1426.97	RC206	3/4	1146.32/1124.37
RC107	11/12	1230.48/1247.37	RC207	3/4	1061.14/1052.91
RC108	10/11	1139.82/1147.04	RC208	3/3	828.14/834.99

Note: VH denotes the number of vehicles used, Dist - total distance

6. Conclusions

A Cooperative Population Learning Algorithm for the Vehicle Routing Problem with Time Windows has been proposed in the paper. It extends original Population Learning Algorithm by incorporating a cooperative problem solving paradigm into it. The process of search for solutions in CPLA is divided into stages where different learning/improvement procedures are used at each stage. These procedures are based on a set of heuristics (represented as software agents) which are run under the cooperation schema defined separately for each stage.

The computational experiment which has been carried out confirmed the effectiveness of the proposed approach. Using a few stages where different learning/improvement procedures are used at each stage allows one to gradually improve solutions obtained in previous stages. As a result, final results can compete with solutions obtained by other heuristic methods dedicated for solving VRPTW.

Acknowledgments.

The research has been supported by the Polish National Science Centre grant no. 2011/01/B/ST6/06986 (2011-2013).

References

- [1] E.G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley and Sons, Inc., 2009.
- [2] F. Glover, M. Laguna, *Tabu Search*, Kluwer, Boston, 1997.
- [3] R.W. Eglese, Simulated annealing: A tool for operational research, *European Journal of Operational Research* **46** (1990), 271–281.
- [4] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, *Journal of Global Optimization* **6** (1995), 109–133.
- [5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin-Heidelberg-New York, 1994.
- [6] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* **11** (1997), 341–359.
- [7] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, Springer, Heidelberg, 2006.
- [8] F. Glover, M. Laguna, R. Marti, Fundamentals of scatter search and path relinking, *Control and Cybernetics* **39** (2000), 653–684.
- [9] M. Dorigo, T. Stutzle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.
- [10] J. Kennedy, R. Eberhart, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks IV* (1995), 1942–1948.
- [11] P. Jędrzejowicz, Social Learning Algorithm as a Tool for Solving Some Difficult Scheduling Problems, *Foundation of Computing and Decision Sciences* **24(2)** (1999), 51–66.
- [12] D. Barbucha, Three approximation algorithms for solving the generalized segregated storage problem, *European Journal of Operational Research* **156** (2004), 54–72.
- [13] J. Jędrzejowicz, P. Jędrzejowicz, Implementation and Experimental Validation of the Population Learning Algorithm Applied to Solving QAP Instances, in: B. Ribeiro, R.F. Albrecht, A. Dobnikar, D. Pearson, N. Steele (eds.), *Adaptive and Natural Computing Algorithms*, Springer, Berlin Heidelberg (2005) 198–202.
- [14] J. Jędrzejowicz, P. Jędrzejowicz, Agent-Based Approach to Solving Difficult Scheduling Problems, in: M. Ali, R. Dapoigny (eds.), *IEA/AIE 2006, LNAI 4031*, Springer, Berlin Heidelberg (2006), 24–33.
- [15] C. Blum, A. Roli, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys* **35(3)** (2003), 268–308.
- [16] S. Talukdar, L. Baerentzen, A. Gove, P. de Souza, Asynchronous Teams: Cooperation Schemes for Autonomous Agents, *Journal of Heuristics* **4(4)** (1998), 295–321.
- [17] D. Barbucha, I. Czarnowski, P. Jędrzejowicz, E. Ratajczak-Ropel, I. Wierzbowska, JABAT Middleware as a Tool for Solving Optimization Problems, *Transactions on Computational Collective Intelligence II, LNCS 6450*, Springer, Berlin Heidelberg (2010), 181–195.
- [18] D. Barbucha, I. Czarnowski, P. Jędrzejowicz, E. Ratajczak-Ropel, I. Wierzbowska, Team of A-teams a Study of the Cooperation between Program Agents Solving Difficult Optimization Problems, in: I. Czarnowski, P. Jędrzejowicz, J. Kacprzyk (eds.), *Agent-Based Optimization, Studies in Computational Intelligence 456*, Springer, Berlin Heidelberg (2013), 123–141.
- [19] O. Brysy, M. Gendreau, Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms, *Transportation Science* **39** (2005), 104–118.
- [20] O. Brysy, M. Gendreau, Vehicle Routing Problem with Time Windows, Part II: Metaheuristics, *Transportation Science* **39** (2005), 119–139.
- [21] M. Solomon, Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research* **35** (1987), 254–265.
- [22] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE. A White Paper, *Exp.* **3(3)** (2003), 6–20.
- [23] TILAB, <http://jade.tilab.com/>
- [24] M. Solomon, VRPTW Benchmark problems, <http://w.cba.neu.edu/~msolomon/problems.htm>

Mobile Agent-based Dynamic Resource Allocation Method for Query Optimization in Data Grid Systems

Igor EPIMAKHOV^a, Abdelkader HAMEURLAIN^a,
Franck MORVAN^a and Shaoyi YIN^{a,1}

^a Institut de Recherche en Informatique de Toulouse IRIT,
Paul Sabatier University, France

Abstract. Resource allocation is one of the principal stages of query processing in relational data grid systems. Specific characteristics of the data grid environment, such as dynamicity, heterogeneity and large scale, impose serious restrictions to the resource allocation process. Static resource allocation before the query execution may be far from optimal due to the dynamic changes of the system. One possible optimization is to adjust dynamically the allocation of resources during the query execution. Some methods of dynamic resource allocation have been proposed, however, most of them use centralized control mechanisms. In this study we argue that the decentralized approach meets better the requirements of the data grid systems. In this study we propose a decentralized method of dynamic resource allocation that is based on the mobile agent paradigm. We consider the participating nodes as autonomous and independent elements of the system, each of which can detect if it is overloaded and make the decision to react. Then we consider each relational operation as a mobile agent running on the allocated node, meaning that, it keeps track of its own status and can migrate to another node at any time. A two-level cooperation mechanism between such autonomous nodes and autonomous operations is described in detail. Performance evaluation proves the efficiency of the proposed method.

Keywords. Data grid systems, distributed query processing, optimization, resource allocation, load balancing, scheduling.

1. Introduction

Resource allocation is one of the principal stages of query processing in relational data grid systems. At a given time, the system containing N nodes receives a query Q and the query is decomposed into M relational operations which are usually dependent tasks. The objective of the resource allocation is to assign a part of the N nodes to perform the M operations such that the response time of Q is minimized. The scientific community gave significant considerations to this problem in the last decade. The dynamic nature of data grids raises four principal problems for the resource allocation: 1) the leaving of nodes forces the system to retransfer their load to the remaining

¹ Corresponding Author: Shaoyi YIN, Institut de Recherche en Informatique de Toulouse IRIT, Paul Sabatier University, 118 Route de Narbonne, 31062 Toulouse, France; Email: shaoyi.yin@irit.fr

nodes; 2) the unpredictable overload of nodes dynamically occurs because of leaving nodes' load retransfer and because of uncoordinated query optimization; 3) the entering nodes should be used quickly in order to balance the load in the system; 4) failures of nodes may cause data loss and data inconsistency, but we leave the fault tolerance problem out of scope of the paper.

In our survey [1] we distinguished two principal strategies for dealing with the dynamic nature of a data grid. The first strategy [2, 3, 4] allocates resources for an operation after finishing the execution of the previous operation. The second strategy [5, 6, 7, 8] combines an initial resource allocation (static phase) with a run-time reallocation (dynamic phase). In the present paper we focus on the dynamic phase of the second strategy.

We analyzed a number of works on the dynamic resource allocation problem and found that most of the published methods rely on the centralized control organization [2, 3, 4, 5, 6, 7, 8, 9, 10]. However, in a large scale dynamic system, the centralized control mechanism has some fundamental shortcomings like low reliability, risk of bottleneck and scaling problem. The main objective of the present study is to propose a decentralized method of dynamic resource allocation that could react to the dynamic fluctuations of a relational data grid system.

In the decentralized design, we consider nodes as autonomous and independent elements of the system, each of which can decide whether to execute an operation or to make it leave. Implemented as mobile agents, operations also have a limited degree of autonomy, and can make decisions about its migration and execution. In order to clearly define such a system, we must resolve the following basic questions:

1. When an operation must migrate?
2. Which operation must migrate?
3. Where should it migrate?

In addition to answering these questions, we must determine an effective structure for the interaction on two levels: 1) between a node and the operations running on it; 2) between the neighboring operations within a query. To treat this problem, we propose a two-level control system that provides cooperation between autonomous nodes and autonomous operations, clearly describing functions of each side.

The rest of the paper is organized as follows. Section 1 presents an analysis and comparison of existing methods and positions our proposed method in the domain. Section 2 describes the core of our method, including the detection of node overload, the determination of the operation to migrate, the selection of the node for migrating operation and the control structure of the system. Section 3 provides the results of performance evaluation and Section 4 concludes the paper.

2. Related Work

The problem of modifying the execution plan during the execution phase in order to adapt to dynamic changes of the grid environment attracts an attention of the scientific community and entails a number of publications in the recent years. Though a large number of works devoted to the issue of resource allocation in general purpose data grids and computational grids, we found just a few studies [2, 3, 4, 5, 6, 7, 8, 10, 11, 12] that concern the problem of dynamic resource allocation in relational data grids.

The main peculiarities of the environment consist of: 1) strong dependencies between tasks that correspond to the physical query operations; 2) high degree of fragmentation and duplication of data in the system.

In our survey [1] we analyzed the domain of resource allocation in data grids. We found that among the studied works there are two principal approaches for the adaptation of query execution plan to the dynamic unstable environment of grid. The first approach is implemented in [2, 3, 4], where the scheduler allocates resources for an operation after finishing the execution of the previous one. In this case, the scheduler can optimize the allocation quality based on the information about the availability and the load of resources, collected during execution of previous operations. The problem with this approach is that its adaptation ability is limited to the moment between finishing one operation and starting the next. The second approach [5, 6, 7, 8] combines an initial resource allocation (static phase) with a continuous execution-time monitoring and reallocation of resources (dynamic phase). The present study adopts the second approach and proposes an execution-time reallocation method.

The mechanisms of dynamic resource allocation differ in objectives, structure of control, type of reaction etc. In the rest of this section, we analyze the differences and try to position our work relative to the existing proposals.

With regard to the objectives, many [5, 6, 7, 8, 9, 10, 13, 14, 15, 16, 17, 18] consider the load balancing problem in order to raise the efficiency of task executions, some of which [6, 7, 8, 10, 13, 15, 16, 17, 18] also have as objective the using of new resources that appear in run-time. Another objective is the fault tolerance [8, 15]. In our method we work on the first two objectives, leaving the last one out of scope of our study.

The control structure of resource reallocation is organized differently by the authors. The work [16] proposes a global hierarchical structure that resolves the load balancing problem on the intra-group, intra-region and intra-grid levels. Many other authors use a locally centralized broker for a query [5, 7, 8, 10]. In the present paper we propose a method where the control structure is completely decentralized.

Concerning the type of reaction to dynamic fluctuations in the system, methods [5, 9] balance the load between nodes by reordering operations in execution time. There are works [17, 18] that, like our method, use a mechanism of migration of operations. Others [7, 8, 10] exploit a query rescheduling performed by a centralized query broker that does not only migrate operations but also can change the level of parallelism dynamically. An important aspect of the operation migration is the detection of the moment to migrate. Most of the authors [13, 14, 15, 16, 17, 18] propose to use the shortage of allocated resources and the presence of idle resources in the system as a triggering event of migration. However, they answer the question of how to detect the shortage of resources differently. In [16] the imbalance state is defined relatively to neighbors by comparing the processing time of a single node with the average processing time of the group containing this node. Some methods [13, 14, 15, 17, 18] detect the shortage of resources by the performance degradation. We define the shortage of resource for each node by comparing its capability with the total resource requirements of all the operations running on it. Another important aspect of the operation migration is the selection of node to which an operation will migrate. Some authors define a set of candidate resources (nodes) for the allocation process. Method [17] tries to maximize the number of dependent tasks that are located to the same node. Methods [16, 19] use neighbor principle to define candidate resources. For example, method [19] identifies neighbors based on the network topology: this approach is rather

precise, but very resource consuming. We propose to combine the data locality and neighbor principles and define a set of candidate resources using the geographical proximity.

3. Dynamic Resource Allocation Method

In this section we discuss the principal elements of our method which resolve the basic questions that we posed above in the Introduction. In our design, the migration of an operation is triggered by two types of events: 1) when a node is overloaded; 2) when a node is leaving. Section 2.1 explains how to detect the overload status of a node. If a node is overloaded, it will iteratively choose some operations and ask them to migrate. Section 2.2 shows how to choose such operations. Regarding the situation of node disconnection, all the operations that are placed on the disconnecting node must be transferred from it for liberating all its resources. When an operation is chosen to migrate, it has to decide which node it will move to. Section 2.3 describes the algorithm of node selection. In the end, Section 2.4 defines a control structure which connects the above elements.

3.1. Detection of Node Overload

To determine the state of overload of a node we need to calculate the requirements of operations that are placed on the node. If the requirements exceed the available resources, then the node is overloaded and it is necessary to free a part of its resources by reducing the number of executing operations.

The total requirements of operations are determined as the sum of requirements of each operation. We calculate separately the requirements of operations for each considered resource (CPU performance R_{CPU} , I/O bandwidth $R_{I/O}$, network connection bandwidth R_{NET}) as follows (Eq. (1)):

$$R_{CPU} = \sum R_i^{CPU}, R_{I/O} = \sum R_i^{I/O}, R_{NET} = \sum R_i^{NET} \quad (1)$$

where R_i^{CPU} , $R_i^{I/O}$ and R_i^{NET} are the requirements in CPU performance, I/O bandwidth and network bandwidth of the i-th operation respectively.

Our criterion of overload is expressed in the equations (Eqs. (2) and (3)):

$$\sum R_i > R_{Total} \quad (2)$$

$$R_i = \{R_i^{CPU}, R_i^{I/O}, R_i^{NET}\}; R_{Total} = \{R_{Total}^{CPU}, R_{Total}^{I/O}, R_{Total}^{NET}\} \quad (3)$$

where R_i is the need of resources for the i-th operation; R_{Total} is the total amount of resources of the node, defined by $R_{Total}^{CPU}, R_{Total}^{I/O}, R_{Total}^{NET}$ that are available CPU performance, I/O bandwidth, network bandwidth of the node respectively.

In general, each operation is working in cooperation with other operations from the same query plan in order to provide requested data to a user. The operations in a query plan are united in a tree, in which the undermost-level operations read initial relations and the upper-level operations sequentially process them to generate the final result on the topmost level. If the operations in different levels are badly synchronized, meaning that the output speed of a sending operation is much higher or much lower than the input capacity of the receiving operation, the overall performance will be degraded.

We propose a method of self-adaption of autonomous operations, in which every operation tries to keep its performance in balance with its neighbors. More precisely, the mobile agent linked to an operation monitors the performance of lower-level neighbors of that operation and calculates the necessary quantity of resources for processing their data. The main idea is that an operation needs such a quantity of resources that it could process the input data with the same speed as the speed of data transfer from the previous operations. By demanding necessary resources from a node, the mobile agents linked to the operations of a query tree optimize and balance the resources in a cascade way from bottom to top.

3.2. Determination of the Operation to Migrate

After the detection of the overloaded state of a node, the system must reduce the load by moving a part of operations to less loaded nodes. We propose an algorithm of task excluding whose objective is to choose an operation from a set of operations to remove from the node. The operation must be transferred to other node only if it allows using optimally the resources of the node and will have a gain in time for the entire set of concurrent operations.

In general, at each moment t a set of operations $O = \{o_1, o_2, \dots, o_n\}$ is executing on the node. This raises the problem of determining the operation o_i from the set O , such that its migration from the current node would decrease the total execution time of the entire set of operations. We propose the following algorithm to determine o_i .

Algorithm of task excluding

INPUT: A set of operations $O = \{o_1, o_2, \dots, o_n\}$

OUTPUT: Operation o_i which should be removed

BEGIN

1. **FOR** each operation $o_i \in O$ **DO**
2. Calculate T_{gain}^i and T_{migr}^i in the case of excluding o_i
3. **ENDFOR**
4. $T_{max} = \max_{i \in O} (T_{gain}^i - T_{migr}^i)$
5. **IF** $T_{max} > 0$ **THEN** return o_i
6. **ELSE** return null value

END

where T_{gain}^i is an estimated gain for the set of operations in the case of removing the i -th operation, T_{migr}^i is an estimation of the time to migrate the i -th operation.

The migration cost of different operations should be evaluated separately. As an example, we study here only the case of Hybrid Hash Join (HHJ) [20], one of the most versatile join algorithms. HHJ can be divided into two stages: 1) in the first stage, each received relation is divided into blocks of tuples, by performing the same hash function; 2) in the second stage, blocks from one relation are joined with the respective blocks from another relation (the join is thus broken up into a series of smaller joins) and the resulting tuples are sent to the recipient node.

Migration in the first stage would actually lead to transferring all the received data and restart the operation at another site, because partially created blocks of relations cannot be reused. We define the cost of migration at this stage as: $T_{migr}(t) = D(t) / S_{local}$ where t is the amount of time that has elapsed since the beginning of the operation, $D(t)$ is the amount of received data at that moment and S_{local} is the local speed of data transferring from the current node.

In the second stage, blocks of tuples are fully generated and can be transferred to another node without having to restart the operation. In fact, due to sorting of the blocks, it is enough to transfer only the remaining unprocessed blocks. We define the migration cost at this stage as: $T_{migr}(t) = (D - N(t)) / S_{local}$ where D is the total size of the joining relations and $N(t)$ is the size of already processed blocks at the time t .

3.3. Selection of the Destination Node

Algorithm of node selection for operation migration analyzes a predetermined set of candidate nodes, which we denote as Migration Space (MS). It may not include all nodes of the data grid and should be limited to the most preferred nodes. For the MS of binary relational operators such as joins, we consider firstly the source nodes of used relations, and secondly nodes that are geographically close to the source nodes (nearest nodes).

We propose an algorithm that selects the node to which the operation can migrate from the current node with minimal loss of the time.

Algorithm of node selection

INPUT: Generated migration space MS with the data about available resources for each node

OUTPUT: Node for the migration of an operation

BEGIN

1. **FOR** each node $i \in MS$ **DO**
2. Calculate $T_{cost}^i = T_{migr2}^i + T_{exec}^i$ for the i -th node
3. **ENDFOR**
4. return the node with $T_{cost} = \min_{i \in MS} (T_{cost}^i)$

END

Where T_{exec}^i is an estimated execution time of all the operations on node i by adding the migrating operation and T_{migr2}^i is an estimation of the time to receive the data of the migrating operation. The estimation of T_{migr2}^i is slightly more precise than the estimation of T_{migr}^i in Section 2.2, because at this step, we have more information about

the destination node, such as the network speed. So the $T_{migr_2}^i$ is defined as follows. If the migration is done during the first phase of the HHJ operation, we have Eq. (4) and if the migration is done during the second phase, we have Eq. (5), where S_{remote} is the data receiving speed of the destination node.

$$T_{migr_2}(t) = D(t) / \min(S_{local}, S_{remote}) \quad (4)$$

$$T_{migr_2}(t) = (D - N(t)) / \min(S_{local}, S_{remote}) \quad (5)$$

3.4. Control Structure for the Dynamic Resource Allocation System

Choosing the paradigm of mobile operators, we have abandoned a central coordinator of the query, which can reduce the reliability of the query execution in a large-scale environment. With the new approach, each operation independently controls its own execution and migrates in cases of node overcharge or disconnection. In the process of the decision making, each operation interacts directly with neighboring operations that play the roles of data sources and data recipients.

We consider the node overload as the main reason for migration. As we defined above, to analyze the load of a node, it is necessary to accumulate information from the entire set of operations that are placed on the node. The node then makes the decision to remove one of the operations from it for the common interests of all operations. To solve this problem, we propose to use a coordinator for each node, giving to it part of responsibility in the decision-making. Thus, in the decision-making process assist two main participants: the node coordinator and the mobile agent (operation). We will describe their authorities, functions and decision-making criteria.

Node coordinator has the authority to make decisions about migration of any operation placed on its node. Its main functions are: 1) collecting information from operations and analyzing it; 2) detecting the node overload status; 3) selecting an operation to remove from the node. As a criterion for decision-making, the node coordinator considers the minimization of the total execution time of all operations placed on it.

The mobile agent makes its own decision when choosing a site on which it will migrate. Its main functions are: 1) exchanging data with neighboring operations; 2) analyzing its own resource requirements; 3) selecting a node for migration. The main decision-making criterion for the mobile agent is to minimize the execution time of all operations on the destination node.

4. Performance Evaluation

We performed an experiment in order to verify the efficiency of our method in a dynamic data grid environment. After analyzing available data grid infrastructures and simulators, we found that none of them meets completely our demands. In a real data grid we cannot use a large part of the system exclusively, so it is not possible to get repeatable results because of significant influence of other tasks, performed by nodes in parallel. Also, we did not find a data grid simulator that can completely simulate an

environment of relational data grid with unstable and heterogeneous nodes, dependent operations of the query, distributed and duplicated relations.

Thus we decided to use for the experiment our own developed simulator, described in [1]. We extended it with the capability to simulate leaving and entering nodes. We implemented also mechanisms of dynamic state monitoring.

4.1. Experiment Conditions and Measured Parameters

We simulated a segment of data grid with 200 nodes that store 200 relations. Each relation contains 5 equal fragments, each of which in turn is duplicated on 4 nodes. So in average we have 20 copies of different fragments on each node. We believe that the chosen scale is sufficient for testing the behavior of our method in dynamic conditions of data grid.

For the experiment we used a set of 100 previously generated queries, each query contains 3 join operations. The system starts the execution of all queries at the same time, so there is a significant load for all components of the system.

Simulator makes an initial resource allocation individually for each query, without taking into account the placement of other queries. That is caused by the absence of a global multi-query scheduler and by the simultaneity of resource allocation processes for the queries. We chose the condition consciously, in order to verify the ability of our method to balance the load.

As measured parameters, we chose the average execution time of a query, the number of migrations during the execution, the average load of I/O subsystem, the average load of local network connections and the number of processing queries at any time. The average execution time shows the query processing efficiency of the method. We measured the average load of I/O subsystems and the average load of network connections for studying the dynamic behavior of the system in execution time. Note that, as the local network connection, we consider the local link that connects a node to a wide-area network which connects all nodes of the data grid system. Measuring the number of queries in process during the execution lets us examine the dynamicity of queries' finalization in different test conditions.

All tests passed with the same initial set of resources and the same set of queries. We tested our system with enabled (*LB*) and disabled (*noLB*) load balancing mechanism. In the disabled mode mobile operations migrate only for liberating the node that is leaving the system. Otherwise they migrate also in order to balance the load and to improve the performance. That permits us to analyze the efficiency of the proposed load balancing method and the influence of the dynamicity of environment to the main parameters of the system.

4.2. Performance Analysis

Hereafter we examine how the main parameters of the system change during the execution in different test conditions.

4.2.1. Average Execution Time

We see in Figure 1 that, comparing to the *noLB* mode, activation of *LB* decreases the average execution time by 39.1%. So we conclude that the proposed method of load balancing significantly increases the efficiency of the system.

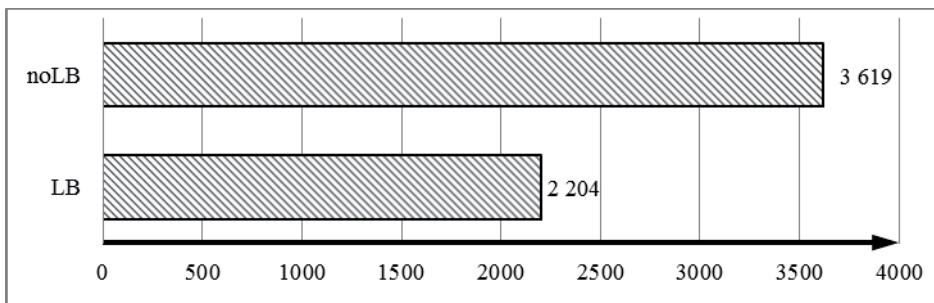


Figure 1. Average execution time of a query (seconds).

4.2.2. Number of Operation Migrations during the Query Execution

25% of the nodes leave the system at the 600th second and reenter the system at the 1200th second, as displayed in Figure 2 by dash lines. The figure shows the number of migrations initiated by the system in different periods of time. As could be seen on it, our method makes a large number of operations to migrate at the beginning. After that, there is a second large peak of migrations, which occurs at the moment of disconnecting of 25% of nodes from the data grid. In our method, all operations allocated to the disconnecting nodes are forced to migrate immediately to other nodes.

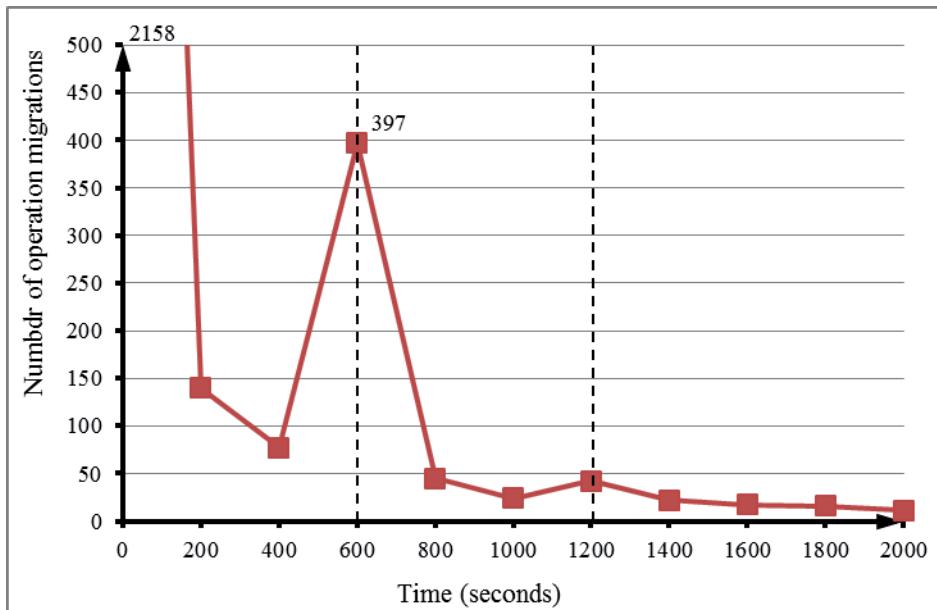


Figure 2. Number of operation migrations during the query execution.

4.2.3. Average Load of I/O Subsystem

The most obvious phenomena in Figure 3 is the large difference between LB enabled and LB disabled modes in terms of duration. The figure shows that the LB method

increases significantly the I/O subsystem utilization, which is consistent with the decreasing of the average execution time.

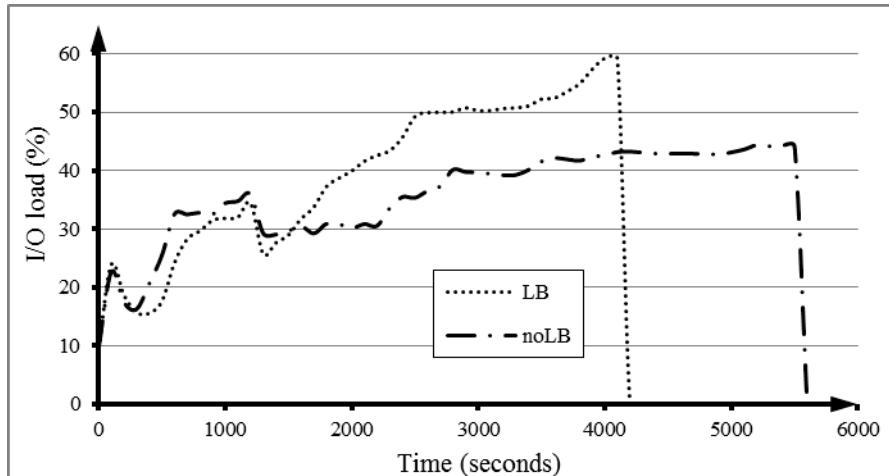


Figure 3. Average load of I/O subsystem.

4.2.4. Average Load of Network Subsystem

We can see in Figure 4 that load balancing makes higher network utilization. Analyzing the results of our experiments, we notice that higher I/O and network utilization brings lower average execution time.

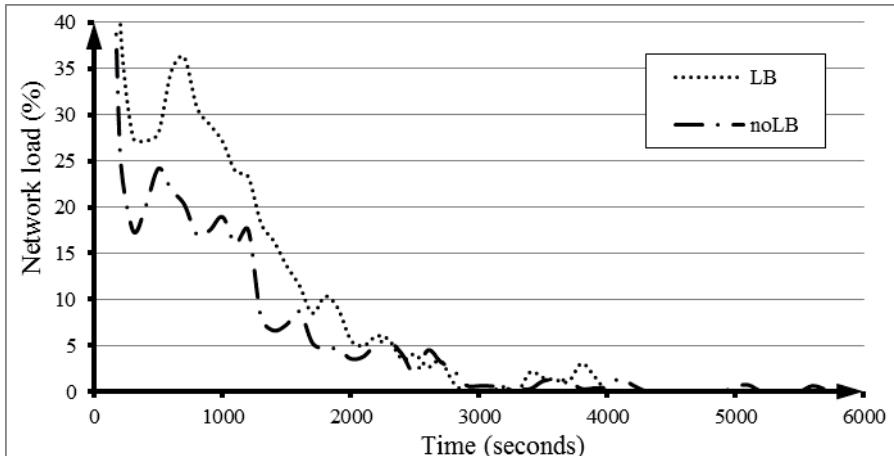


Figure 4. Average load of network subsystem.

4.2.5. Number of Queries in Process

The last diagram in Figure 5 demonstrates the dynamicity of the finalization of queries in different testing modes. The mode with enabled load balancing starts first finalizing

a bit earlier than the mode without it. Then the difference grows and becomes more significant in the end of execution than at the beginning.

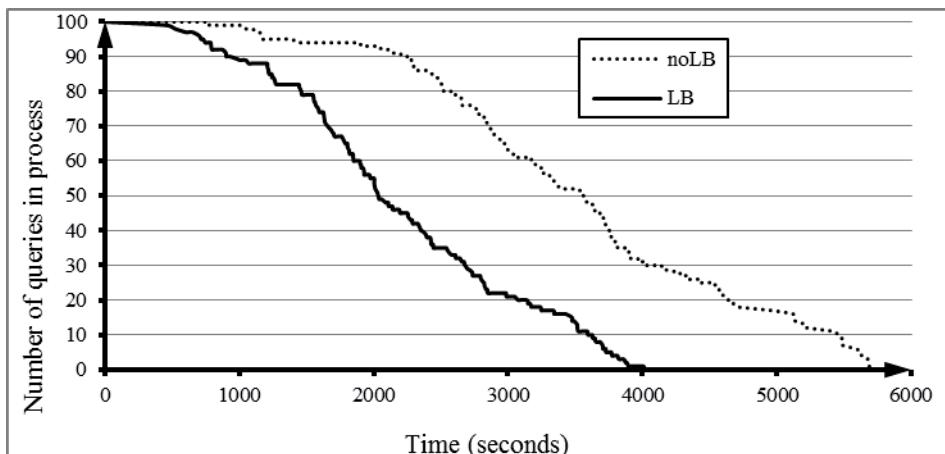


Figure 5. Number of queries in process during the query execution.

5. Conclusion

In this paper we presented a dynamic resource allocation mechanism for data grid systems. The proposed method adopts a decentralized approach, where each computing node is autonomous and can make the decision to remove operations from it. We use a mobile agent paradigm, where each operation is implemented as a mobile agent and can autonomously migrate to another node. We described the algorithms of node overload detection, migrating operation determination and destination node selection. We designed a two-level decentralized control system, which allows the cooperation between autonomous nodes and autonomous operations.

The results of performance evaluation prove the efficiency of the proposed method in terms of load balancing and nodes' instability tolerance. Comparing results with and without load balancing, we see that the proposed method increases average load of I/O and network subsystems and therefore decreases the average query execution time by 39.1%.

We conclude that the dynamic resource allocation phase during the query execution decreases efficiently the query response time in data grid systems. We believe that the decentralization of control combining with the mobile agent paradigm is a very promising approach. As the future work, we will compare our method with other related work by doing more experiments.

Acknowledgement

This work was supported in part by the French National Research Agency ANR, PAIRSE Project, Grant number -09-SEGI-008.

References

- [1] I. Epimakhov, et al. Resource scheduling methods for query optimization in data grid systems. *Advances in Databases and Information Systems*, volume 6909 of *Lecture Notes in Computer Science*, 185–199. Springer Berlin / Heidelberg, 2011.
- [2] H. Chen and Z. Wu. Dartgrid iii: A semantic grid toolkit for data integration. In *Proceedings of the First International Conference on Semantics, Knowledge and Grid (SKG '05)*, 12–, Washington DC, 2005.
- [3] H. Chen, et al. Dartgrid: a semantic infrastructure for building database grid applications: Research articles. *Concurrency Computation : Practice & Experience*, 18(14):1811–1828, December 2006.
- [4] Z. Wu, et al. Dartgrid: Semantic-based database grid. In *Proceedings of the 4th International Conference on Computational Science*, 59–66, Krakow, Poland, June 2004.
- [5] V. F. V. Da Silva et al. An adaptive parallel query processing middleware for the grid. *Concurrency and Computation: Practice and Experience*, 18(6):621–634, 2006.
- [6] A. Gounaris, et al. Modular adaptive query processing for service-based grids. *CoreGRID Technichal Report Number TR-0076*, 2007.
- [7] A. Gounaris, et al. Adaptive query processing and the grid: Opportunities and challenges. In *DEXA Workshops*, 506–510, 2004.
- [8] A. Gounaris, et al. Practical adaptation to changing resources in grid query processing. In *Proceedings of the 22nd International Conference on Data Engineering*, ICDE '06, pages 165–168, Washington DC, 2006.
- [9] R. Avnur and J. M. Hellerstein. Eddies: Continuously adaptive query processing. In *Proceedings of the SIGMOD Conference*, 261–272, 2000.
- [10] A. Gounaris, et al. Self-monitoring query execution for adaptive query processing. *Data Knowl. Eng.*, 51:325–348, December 2004.
- [11] D. Cokuslu, et al. Resource allocation for query processing in grid systems: a survey. *Computer Systems: Science& Engineering*, 27(4), 2012.
- [12] A. Gounaris, et al. Adaptive query processing: A survey. In *Proceedings of the 19th British National Conference on Databases (BNCOD)*, 11–25, 2002.
- [13] R. Al-ali, et al. Qos adaptation in service-oriented grids. In *Proceedings of the 1st International Workshop on Middleware for Grid Computing (MGC2003) at ACM/IFIP/USENIX Middleware 2003*, Rio de Janeiro, 2003.
- [14] R. Buyya, et al. A case for economy grid architecture for service oriented grid computing. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, 776 –790, April 2001.
- [15] E. Huedo, et al. An experimental framework for executing applications in dynamic grid environments. *ICASE Technical Report No. 2002-43*, 2002.
- [16] J. Patni, et al. Load balancing strategies for grid computing. In *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT)*, 239 –243, April 2011.
- [17] X. Sun and M. Wu. Ghs: a performance system of grid computing. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, Chicago, April 2005.
- [18] S. Vadhiyar and J. Dongarra. A performance oriented migration framework for the grid. In *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003)*. 130 –137, May 2003.
- [19] D. Cokuslu, et al. Resource allocation algorithm for a relational join operator in grid systems. In *Proceedings of the 16th International Database Engineering & Applications Symposium (IDEAS '12)*, 139–145, New York, 2012.
- [20] D. A. Schneider and D. J. DeWitt. A performance evaluation of four parallel join algorithms in a shared-nothing multiprocessor environment. *SIGMOD Record*, 18(2):110–121, June 1989.

Memetic Multi-Agent Computing in Difficult Continuous Optimisation

Aleksander BYRSKI^{a,1}, Wojciech KORCZYŃSKI^a and
Marek KISIEL-DOROHINICKI^a

^aAGH University of Science and Technology, Krakow, Poland

Abstract. In the paper an application of hybridized Evolutionary Multi-Agent System (EMAS) with local search (in memetic style) to the problem of continuous optimisation is presented. Before, the concept of evolutionary and memetic agent-based computing is given, the former being a computing paradigm researched for over 15 years, the latter being introduced recently. Two ways of memetic hybridization (Lamarckian and Baldwinian) are discussed, and examined in the course of experiments. In the presented experiments, evolutionary and memetic multi-agent systems are compared with classical evolutionary algorithm (Michalewicz model) implemented with allopatric speciation (island-model of evolutionary algorithm), based on a selected popular benchmark continuous optimization functions.

Keywords. agent-based computing, memetic-computing, global-optimisation, meta-heuristics

1. Introduction

The reason why some problems are perceived as “difficult” for optimization methods is because their domains are very hard or even impossible to be described and explored, using conventional mathematical apparatuses (see, e.g. combinatorial optimization problems [1]). Such “black-box” problems [2] may be solved only using a general-purpose algorithms, such as meta-heuristics, taking into consideration little, if any information from a problem domain, to devise a solution.

The success story of meta-heuristics is marked by the increasing understanding of them, being a true mark of heuristic methods—working but not fully understood in advance, until the experimental research is conducted. One of the important achievements was made by early practitioners such as L. Davis [3] and P. Moscato [4], namely the need for adjusting the solver to the problem by exploiting knowledge available on the latter. Initially, this practical observations were theoretically backed up by [5] and later by [6] in the *No-Free-Lunch Theorem*. The realization of this achievement constitutes one of the raisons-d’être of optimization techniques such as memetic algorithms (MAs) [7,8,9]. These are population-based techniques that blend together ideas from other meta-heuristics, most commonly in terms of integrating local search within the population-based search engine. This definition of MA was actually popularized by early works

¹Corresponding Author: Aleksander Byrski, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland; E-mail: olekb@agh.edu.pl.

such as [10] and paved the way for the vigorous development of optimization algorithms based on this idea, exhibiting a remarkable record of success, check, e.g., [11].

Coming back to the roots of the *memes theory* [7] the article concerns a hybrid evolutionary agent-based computing (so called *evolutionary multi-agent system* – EMAS [12]) enhanced with local search algorithms, making together a truly agent-based memetic computing system.

In the course of paper, after giving the basics of evolutionary and memetic agent-based computing and presenting the concepts of the examined systems, the experimental results concerning optimisation of a high-dimensional benchmark function are given and discussed, and in the end, the conclusions are presented.

2. Memetic Population-based Metaheuristics

Population-based meta-heuristics, in particular evolutionary algorithms [13] are universal optimisation techniques. Instead of directly solving the given problem, the problem is encoded in a special way (genotype) and random populations of potential solutions are constructed. Based on the existing fitness function (evaluating the genotype), selection is performed (so the mating pool is created) and based on the mating pool, the subsequent population is created with use of predefined variation operators (such as crossover and mutation). The process continues until some stopping condition is reached (e.g., number of generations, lack of changes in the best solution found so far). This kind of search has some drawbacks (as possibility of premature convergence), thus several techniques, as multi-deme approaches [14] are applied.

Evolutionary algorithms may be further enhanced, by hybridisation with local-search methods. Such techniques are nowadays called memetic algorithms [7]. In these algorithms, two kinds of search-enhancements are usually implemented:

- Baldwinian local search—based on Baldwin theory stating that predispositions may be inherited during reproduction, implemented usually as hybridisation of local search in the course of evaluation process in. The evaluated individual receives the fitness function value computed for one of its possible descendants (effects of local-search starting from this individual).
- Lamarckian local search—based on Lamarck theory stating that characteristics of individuals acquired in the course of life may be inherited by their descendants, implemented usually as hybridisation of mutation operator. The search for a mutated individual is based not only stochastic one-time sampling from the solution space, it may be a much more complex process, being an outcome of a local search starting from this individual.

Although both theories turned-out to be false, the meta-heuristics based on them are effective in many problems (see, e.g., [15,16]).

The schematic presentation of evolutionary algorithm used as a reference in this paper is presented in Figure 1(a). Of course hybridisation of this algorithm is performed according to above-described rules, yielding Baldwinian or Lamarckian operators (for evaluation and mutation). It may be seen, that the population of potential (encoded) solutions of a given problem is decomposed into evolutionary islands (there is also a possibility of migration between them). The most important fact is that the evolutionary algorithm is

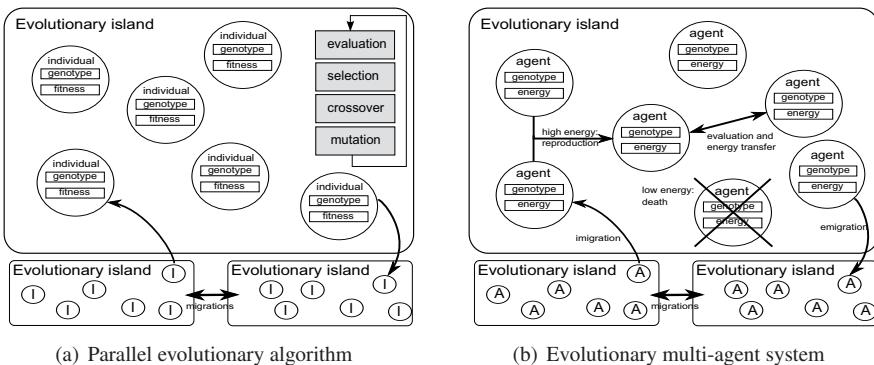


Figure 1. Schematic presentation of biologically-inspired algorithms used in this paper (PEA and EMAS)

common to all islands, all operators are applied one by one, during each of generations, to all parts of the population. After meeting some kind of stopping condition, the best solution so far is presented as the optimal one. One of the main drawbacks of such an approach is global (god-like) selection algorithm - possibilities of its de-globalisation will be described later.

3. Memetic Agent-based Computing

In evolutionary multi-agent systems, an agent represents solutions for a given problem. Core properties of the agent are encoded in its genotype and inherited from its parent(s) with the use of mutation and recombination operators. Besides, an agent may possess some knowledge acquired during its life, which is not inherited. Both inherited and acquired information determines the behaviour of an agent in the system (phenotype). Assuming that no global knowledge is available and autonomy of the agents, selection is based on non-renewable resource, most often called *life energy* [12]. Thus a decisive factor of the agent's activity is its fitness, expressed by the amount of energy it possesses. The agent gains energy as a reward for 'good' behaviour, and loses energy as a consequence of 'bad' behaviour. Selection is realised in such a way that agents with high energy level are more likely to reproduce, while low energy increases the possibility of death. The agents are located on islands, which constitute their local environment where direct interactions may take place, and represent a distributed structure of computation. Obviously, agents are able to change their location, which allows for diffusion of information and resources all over the system.

EMAS agents may perform the following actions (see Fig. 1(b)):

- *Reproduction* – performed when the agent's energy raises above a certain level, followed by production of a new individual in cooperation with one of its neighbours, with genotype based on parents' genotypes (crossed over and mutated) and part of energy (usually half of its initial value) also passed from each of its parents.
- *Death* – agent is removed from the system when its energy falls below a certain level, the remaining energy is distributed among its neighbours.

- *Evaluation* – agent chooses its neighbour and compares the fitness of its genotype with its own; in the case when the neighbour is better, it receives part of the agent's energy, and vice versa.
- *Migration* – agent (with some probability) may migrate, then it is removed from one evolutionary island and moved to another (random) according to predefined topology.

Each action is attempted randomly with certain probability, and it is performed only when their basic preconditions are met (e.g. an agent may attempt to perform the action of reproduction, but it will reproduce only if its energy rises above certain level and it meets an appropriate neighbour).

EMAS computing has already been approached for different problems, e.g. evolution of neural network architecture [17]. Moreover, dedicated models have been constructed in order to verify this approach [18,19,20].

Implementation of Baldwinian and Lamarckian memetics in EMAS is carried out in the following way:

- Baldwinian memetics: this implementation is done in a similar way as in classical evolutionary computing: the evaluation operator is enhanced with local search algorithm. The evaluation of a certain individual starts the local search from this individual and returns the fitness of the solution found instead of the original fitness value.
- Lamarckian memetics: a dedicated mutation operator is called in the course of agent's life, therefore its genotype may be changed whenever this action is undertaken.

4. Experimental Results

Having experience in development of component-based agent-oriented computing platforms (cf. AgE²), and different aspects of design and implementation (e.g., tackling functional integrity of such systems [21]), a simplified version of such discrete-event simulation and computing system was developed using Python technology. The choice of this technology was undertaken based on a relatively easy implementation process and high portability. Using this software environment, both EMAS (evolutionary multi-agent system) and PEA (parallel evolutionary algorithm) systems were implemented based on Python technology and used to generate the presented results. These results were gathered using a server-class hardware (SUN FIRE X2100: Dual-Core AMD Opteron(tm) Processor 1220 2.8GHz, 4GB RAM (2 x 2GB), 1 x 250GB SATA). The configuration of the both systems is presented as follows.

- Common parameters: normal distribution-based mutation of one randomly chosen gene, single-point crossover, the descendant gets parts of its parents genotype after dividing them in one randomly chosen point, 30 individuals located on each island, all experiments were repeated 10 times and standard deviation was computed. allopatric speciation (island model), fully connected islands, 3000 steps of experiment, genotype of length 50, agent/individual migration probability 0.01.

²<http://age.iisg.agh.edu.pl>

- PEA-only parameters: mating pool size equals to the number of individuals, individuals migrate independently (to different islands).
- EMAS-only parameters: initial energy: 100, received by the agents in the beginning of their lives, minimal reproduction energy: 90, required to reproduce, evaluation energy win/loose: 40/–40, passed from the looser to the winner, death energy level: 0, used to decide which agent should be removed from the system, boundary condition for the intra-island lattice: fixed, the agents cannot cross the borders, intra-island neighbourhood: Moore's, each agent's neighbourhood consists of 8 surrounding cells, size of 2-dimensional lattice as an environment: 10×10 , all agents that decided to emigrate from one island, will immigrate to another island together (the same for all of them).

Besides observation of the best fitness according to the step of computation, the diversity was also performed using two selected definitions of this measure:

- Morrison-De Jong (MOI) measure based on concept of moment of inertia for measurement of mass distribution into arbitrarily high dimensionality spaces [22]
- maximum standard deviation (MSD) of each gene computed for all individuals in the population

Memetic operators were implemented according to gradient-free steepest descent algorithm based on choosing the best from 10 potential mutated individuals. Such a procedure was repeated 10 times and the best result was returned. The exact local search algorithms were implemented according to the two following strategies:

- Isotropic mutation – it is a method aimed at generating uniform sampling points on and within N-dimensional hyperspheres. The idea of the Isotropic method algorithm is as follows: firstly the N normal distributed numbers z_i are generated. Then the vectors x are computed by making a projection onto surface by dividing each generated number z_i by $r = \sqrt{\sum_{i=1}^N z_i^2}$. Since the z vectors are isotropically distributed, the vectors x will be of norm 1 and also isotropically distributed. Therefore the points will be distributed uniform of the hypersphere. The generation of points inside the hypersphere may be achieved by rescaling the coordinates obtained in the previous steps. While rescaling, the dimension must be taken into consideration [23].
- Solis and Wets' algorithm – it is a randomised version of optimization technique belonging to a hill climbing family. Every step size can be adapted very quickly as follows: it starts at a current point \vec{x} and checks if either $\vec{x} + \vec{d}$ or $\vec{x} - \vec{d}$ is better, where \vec{d} is a deviate chosen from a normal distribution whose standard deviation is given by a parametr ρ . If the answer is positive then a move to the better point is made and a success is recorded. Otherwise, a failure is recorded. If several successes in a row happen, ρ parameter is increased to make moves more quickly. However if several failures in a row are recorded, ρ is decreased to focus the search. Additionally, a *bias* parameter is used to direct the search momentum in areas that yield success [24].

In order to compare the time-cost of the researched algorithms, the length of computing of particular steps gathered for the researched systems is shown in Table 1. At a first glance, computing with EMAS bears higher time cost than with PEA, however one must

Table 1. Execution time of steps

System	Avg. time [ms]	Std. dev	Std. dev %
EMAS	82.46	25.43	30
EMAS + Lamarck	138.48	45.03	32
EMAS + Baldwin	120.84	34.02	28
PEA	75.63	6.05	7
PEA + Lamarck	487.84	127.48	26
PEA + Baldwin	79.41	10.71	13

remember, that these results were gathered for the case of optimisation of simple benchmark function, and the total execution time will surely be much higher for PEA than for EMAS in the case of complex fitness function (cf. the paragraph describing the average number of fitness function calls in the current section for the all researched systems). Moreover, these results clearly show that computing with EMAS is more unpredictable than in PEA, as the standard deviations are higher in this case.

In Fig. 2, graphical comparison of EMAS and PEA computations were shown. All the graphs show the averaged experimental results along with standard deviation to visualize the dispersion of the experiments.

- In Figs. 2(a) and 2(b), the best fitness in each generation was shown for PEA and EMAS. Moreover, the variations of both systems using isotropic memetics were presented. It seems that in all cases EMAS turned out to be better than PEA, firstly, because it achieved a better result, and secondly, because PEA appears to be stuck in a local extremum (starting from about 500th generation), while EMAS continues to improve the result.
- In Figs. 2(c), 2(d), 2(e), 2(f) the diversity measures for PEA and EMAS with and without memetics are presented. Both PEA and EMAS diversities are similar in the end of experiment, however it is easy to see, that the diversity of EMAS decreases slower than it may be observed for PEA.
- In Figs. 2(g) and 2(h) comparison of the application of two memetic techniques (isotropic mutation and Solis Wets operator) are shown. Again, EMAS turns out to be a better algorithm, moreover, especially in the Solis Wets case.

General observation of all the experiments yields that Baldwin memetics are not effective in the examined cases. It seems, that in optimisation of functions with many local extrema, Lamarckian mutation after localising of a better solution (in a neighbouring local extremum), moves the search near this solution (as the mutation operator changes the genotype). In Baldwinian case, the solutions, from which the neighbouring local extremum was reached, are of course preferred, but the Lamarckian “step” is not performed. Therefore, virtually in all cases the effects of this operator are similar to the ones achieved with the use of evolution.

In Table 2 the summary of the obtained results for computing of several chosen benchmarks [25] is presented. All the results shown, were obtained in 3000th step of computation. Besides giving best fitness, diversity computed according to two applied methods (MSD and MOI) was presented. It is easy to see, that EMAS is always at least a little better than PEA when examining the results, although the dispersion is sometimes bigger and diversity is lower.

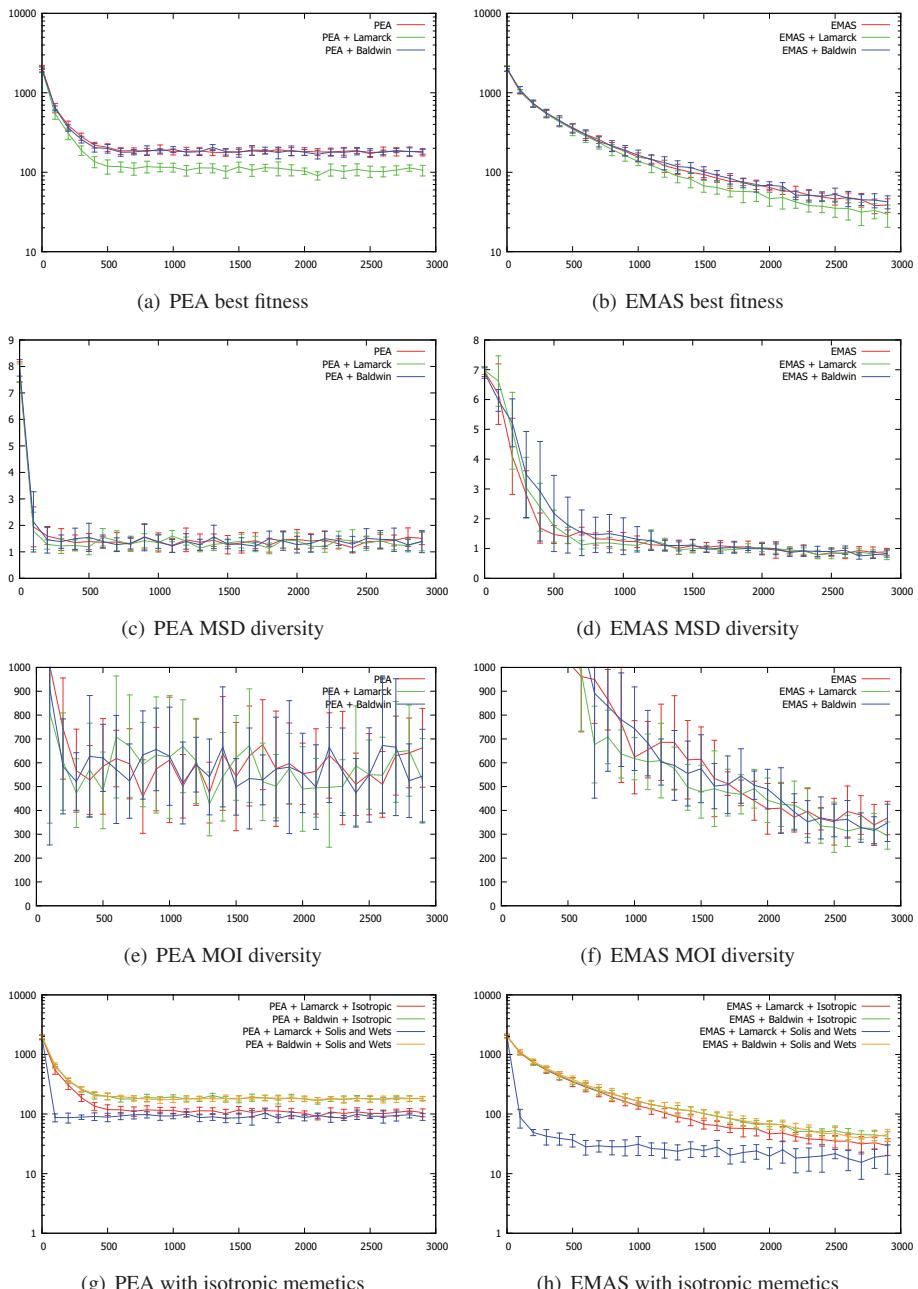


Figure 2. Comparison of results of PEA and EMAS computations (Rastrigin function)

Table 2. EMAS and PEA optimization results obtained for selected benchmark functions [25,26]

	Result	St. Dev.	MSD div.	MSD st. dev.	MOI div.	MOI st. dev.
Ackley						
PEA	2.08	0.13	0.68	0.13	158.97	18.99
EMAS	0.53	0.12	0.29	0.02	44.71	6.54
PEA + Lamarck	1.47	0.16	0.65	0.14	182.21	21.13
EMAS + Lamarck	0.55	0.16	0.31	0.02	40.26	5.32
PEA + Baldwin	2.00	0.11	0.74	0.17	203.60	48.53
EMAS + Baldwin	0.59	0.15	0.32	0.03	48.02	4.30
DeJong						
PEA	7.09	1.08	0.78	0.15	254.13	47.30
EMAS	0.90	0.82	0.54	0.13	70.83	13.53
PEA + Lamarck	4.88	1.32	0.80	0.10	251.11	37.54
EMAS + Lamarck	0.57	0.59	0.53	0.12	75.06	13.19
PEA + Baldwin	6.94	0.55	0.79	0.13	242.98	30.43
EMAS + Baldwin	0.90	0.87	0.53	0.09	79.13	11.83
Rosenbrock						
PEA	1236.51	306.99	0.71	0.10	270.02	56.29
EMAS	197.49	148.09	0.53	0.15	81.71	25.86
PEA + Lamarck	747.56	304.89	0.70	0.07	241.90	26.81
EMAS + Lamarck	192.80	159.11	0.45	0.08	66.62	12.85
PEA + Baldwin	1306.39	246.62	0.84	0.13	292.95	73.28
EMAS + Baldwin	319.13	331.79	0.49	0.07	71.80	9.74
Axis Parallel Hyper Ellipsoid						
PEA	162.17	34.94	1.13	0.46	310.53	97.71
EMAS	17.08	16.63	0.55	0.11	92.38	21.69
PEA + Lamarck	79.90	22.06	1.15	0.49	305.70	74.34
EMAS + Lamarck	13.67	22.26	0.56	0.09	79.64	15.21
PEA + Baldwin	146.41	25.57	1.10	0.28	332.08	83.26
EMAS + Baldwin	16.48	17.68	0.60	0.11	93.35	15.76
Moved Axis Parallel Hyper Ellipsoid						
PEA	822.36	165.32	1.00	0.16	297.72	70.83
EMAS	40.51	23.08	0.53	0.11	85.95	22.92
PEA + Lamarck	405.54	143.66	0.93	0.19	279.10	69.13
EMAS + Lamarck	43.54	43.26	0.56	0.08	74.41	16.75
PEA + Baldwin	745.73	169.04	1.10	0.35	323.41	84.48
EMAS + Baldwin	99.99	58.52	0.53	0.10	80.01	15.07

5. Conclusions

Memetic computing brings new possibilities in hybridizing evolutionary computing approaches. In this paper, the application of memetic operator to enhancing agent-based evolutionary computing was shown. A series of experiments was performed in the case of continuous optimisation, for selected benchmark function. The additional difficulty of the benchmarks was ensured by using their high dimensional versions (50 dimensions in this paper, while other experiments yielded successful results for even higher dimensions: 100 and even 1000). The obtained results confirmed, that EMAS outperforms PEA

in all examined cases. Moreover, EMAS coupled with memetics is even better. In the future work, further testing is planned, including covering broader range of static and dynamic benchmark functions, as well as complex practical combinatorial problems (such as Job Shop or 1-dimensional bin packing).

Acknowledgment

The research presented here was partially supported by the grant “Biologically inspired mechanisms in planning and management of dynamic environments” funded by the Polish National Science Centre, No. N N516 500039.

References

- [1] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Inc., 1998.
- [2] Stefan Droste, Thomas Jansen, and Ingo Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39:525–544, 2006.
- [3] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold Computer Library, New York, 1991.
- [4] P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
- [5] W.E. Hart and R.K. Belew. Optimizing an arbitrary function is hard for the genetic algorithm. In R.K. Belew and L.B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 190–195, San Mateo CA, 1991. Morgan Kaufmann.
- [6] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
- [7] Pablo Moscato. Memetic algorithms: a short introduction. In *New ideas in optimization*, pages 219–234, Maidenhead, UK, England, 1999. McGraw-Hill Ltd., UK.
- [8] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.
- [9] P. Moscato and C. Cotta. A modern introduction to memetic algorithms. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research and Management Science*, pages 141–183. Springer, 2 edition, 2010.
- [10] N.J. Radcliffe and P.D. Surry. Formal Memetic Algorithms. In T. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, volume 865 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin, 1994.
- [11] W.E. Hart, N. Krasnogor, and J.E. Smith. Memetic evolutionary algorithms. In *Recent advances in memetic algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 3–27. Springer-Verlag, 2005.
- [12] K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki. The application of evolution process in multi-agent world (MAW) to the prediction system. In M. Tokoro, editor, *Proc. of the 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96)*. AAAI Press, 1996.
- [13] Z. Michalewicz. *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994.
- [14] E. Cantú-Paz. A summary of research on parallel genetic algorithms. *IlliGAL Report No. 95007. University of Illinois*, 1995.
- [15] K.W.C Ku and M.W. Mak. Exploring the effects of lamarckian and baldwinian learning in evolving recurrent neural networks. In *Proc. of 1997 IEEE Int. Conf. on Evolutionary Computation*. IEEE, 1997.
- [16] D. Whitley, V. Scott Gordon, and K. Mathias. Lamarckian evolution, the baldwin effect and function optimization. In Y. Davidor, Schwefel H.-P., and Männer R., editors, *Proc. of Parallel Problem Solving from Nature III*. Springer, 1994.

- [17] A. Byrski and M. Kisiel-Dorohinicki. Immune-based optimization of predicting neural networks. In V. Sunderam, D. Van Albada, and P. Sloot, editors, *Proc. of Int. Conference on Computational Science (ICCS 2005), Atlanta, GA, 22-25 May 2005*, pages 703–710. Springer, LNCS 3516, 2005.
- [18] Aleksander Byrski and Robert Schaefer. Stochastic model of evolutionary and immunological multi-agent systems: Mutually exclusive actions. *Fundamenta Informaticae*, 95(2-3):263–285, 2009.
- [19] Robert Schaefer, Aleksander Byrski, and Maciej Smolka. Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions. *Fundamenta Informaticae*, 95(2-3):325–348, 2009.
- [20] A. Byrski and R. Schaefer. Formal model for agent-based asynchronous evolutionary computation. In *Proc. of IEEE Congress on Evolutionary Computation Location: Trondheim, NORWAY Date: MAY 18-21, 2009*. IEEE, 2009.
- [21] K. Cetnarowicz and R. Dreewski. Maintaining functional integrity in multi-agent systems for resource allocation. *Computing and Informatics*, 29(6):947–973, 2010.
- [22] Ronald W. Morrison and Kenneth A. De Jong. Measurement of population diversity. In P. Collet et al., editor, *Proc. of EA 2001, LNCS 2310*, pages 31–41. Springer, 2002.
- [23] M. Luban and L. P. Staunton. An efficient method for generating a uniform distribution of points within a hypersphere. *Computers in Physics*, 2:55–60, November 1988.
- [24] F.J. Solis and R.J. Wets. Minimization by random search techniques. *Mathematical Methods of Operations Research*, 6:19–30, 1981.
- [25] Hartmut Pohlheim. Geatbx: Example functions. <http://www.geatbx.com/docu/fcnindex-01.html>, accessed November 2012.
- [26] J. Digalakis and K. Margaritis. An experimental study of benchmarking functions for evolutionary algorithms. *International Journal of Computer Mathematics*, 79(4):403–416, April 2002.

Web Services and Semantic Web

A Technical Survey for Linked Open Data Federation Building

Cuong Duc NGUYEN ¹, Trong Hai DUONG

*School of Computer Science and Engineering,
International University - VNU - HCM, Vietnam*

Abstract. Linked Open Data (LOD) is a successfully initiative Semantic Web, which uses URIs and RDF techniques to connect related pieces of data on the Semantic Web. A rapidly growing number of LODs leads to new challenges for query processing. SPARQL-endpoint has been proposed to access LODs through SPARQL language, however, it requires a holistic understanding of the data when making a SPARQL query and supports a limited number of related LODs. To access distributed LODs, the LOD-federation is proposed. The LOD-federation is an abstract layer consisting a federated ontology and mechanisms for user query processing across LODs via the corresponding endpoints. In this study, a technical survey of recent works supporting for the LOD-federation building, is presented. An effective method of LOD ontology integration is suggested in order to build the federated ontology. Query decomposition algorithm is depicted to translate a query to the LOD-federation into subsequent LODs. Result integration method using co-reference entity and join operations, is also presented.

Keywords. Linked Open Data, SPARQL-endpoint, Ontology Integration, Semantic Web, Linked Open Data Federation

1. Introduction

The Web is rapidly increased and considered as a global information space with linked documents. The huge duplicated information is available in the Internet. However, it is not easy for consumers to look for the information need since lack of semantic, which leads to ambiguous problem in information retrieval and integration. In 2001, Tim Berners-Lee and colleagues have established a vision of the Semantic Web [1] - an extended Web of machine-readable information and automated services that extend far beyond current capabilities. Linked Open Data (LOD) is a successfully initiative today's Semantic Web, which uses URIs and RDF techniques to connect related pieces of data on the Semantic Web. The Linked Data principles provide a basis for realizing this Web of Data, or Seman-

¹Corresponding Author: Cuong Duc Nguyen, School of Computer Science and Engineering, International University - VNU - HCM, Quarter 6, Linh Trung w., Thu Duc dist., HCMC, Vietnam; E-mail: ndcuong@hcmiu.edu.vn.

tic Web. A rapidly growing number of LODs leads to new challenges for query processing. SPARQL-endpoint has been proposed to access LODs using SPARQL language. The disadvantages of a SPARQL-endpoint are the requirement of holistic understanding of the data when making a query and one SPARQL-endpoint only supports for a very limited number of LODs. The federation techniques are becoming a more and more important, since the information storing and processing systems become larger and more distributed. Three different approaches of federation and integration of linked data sources were distinguished in [2], including integration in a central repository, federation over multiple single repositories, and federation over multiple SPARQL endpoints. The last one, called LOD federation, is mainly focused on this work. The LOD federation is considered as an abstract level above SPARQL endpoints, includes a federated ontology and mechanisms for user query processing across the multiple SPARQL endpoints.

In this paper, a technical survey of recent works supporting for the LOD-federation building, is presented. The LOD-federation is an abstract layer consisting a federated ontology and mechanisms for user query processing across LODs. An effective method of LOD ontology integration is introduced in order to build federated ontology. Query transformation algorithm is depicted to translate a query to the LOD-federation into subsequent LODs. Result integration method using co-reference entity and join operations, is also presented.

2. Related Works

Federation is initially proposed for query optimization in distributed databases where many databases have the same schema and the data is duplicated, and federated databases where the schema may be different between instances, but there is a global system to obtain any necessary data. The federation for integrating multiple data warehouses, called data warehouse federation, was distinguished with federation for databases in [3]. Recently, federation has been considered for integrating diverse LOD sources, called LOD federation, where consists of a global ontology and mechanisms for user query processing. In fact, the technologies for federated database building are similar to the ones in the federated Linked Data scenario including a global schema, query transformation, and result integration. However, there are some major differences between federated database and LOD federation. Optimization Strategies were discussed for federated or distributed databases in [4,5,6,7], which rely on the cooperation of the individual databases, since the wrappers are typically used to abstract from diverse schema in different database sources. Query processing in the LOD federation can not be optimized in the same way since the wrappers are not used in the LOD federation and the SPARQL protocol only defines how a query are evaluated by the endpoints (third party providers) [8]. In particular, Schwarte et al. proposed query optimization techniques for LOD federation including new join processing strategies such as joins ordering and bound joins to minimize the number of requests sent to subsequent LOD components, exclusive group mechanism to group triple patterns into a single query that can be exclusively evaluated by single endpoints, and an effective approach for source selection without the need of preprocessed metadata

[9]. The techniques are being implemented to provide an efficient solution for distributed query processing across LOD sources in FedX² [10]. FedX is incorporated into Sesame as a SAIL (Storage and Inference Layer), which is Sesame's mechanism for allowing seamless integration of standard and customized RDF repositories. The underlying Sesame infrastructure enables heterogeneous data sources to be used as endpoints within the federation. FedX implements the logics for optimization and efficient execution of the query in the distributed setting.

3. Basic Notions

To access to information organizing by an ontology - based LOD via a SPARQL-endpoint, the ontology should be provided for agents to be understandable its corresponding LOD. A SPARQL should be created based on the ontology. We assume a real world (\mathbf{A}, \mathbf{V}) where \mathbf{A} is a finite set of attributes and \mathbf{V} is the domain of \mathbf{A} . Also, \mathbf{V} can be expressed as a set of attribute values, and $\mathbf{V} = \bigcup_{a \in \mathbf{A}} V_a$ where V_a is the domain of attribute a . In this chapter, we accept the following assumptions [11,12]:

Definition 1 (Ontology). *An ontology is a triplet:*

$$O = (C, \sum, R) \quad (1)$$

where,

- C : a set of concepts (the classes).
- R : a set of binary relations between the concepts from C .
- $\langle C, \sum \rangle$: the taxonomic structure of the concepts from C where \sum is the collection of subsumption (\sqsubseteq), equivalence (\equiv), and disjointness (\sqcap) relationships between two concepts from C .

Definition 2 (Concept). *A concept c of an (\mathbf{A}, \mathbf{V}) -based ontology is defined as a pair:*

$$c = (A^c, V^c) \quad (2)$$

where c is the unique identifier for instances of the concept. $A^c \subseteq \mathbf{A}$ is a set of attributes describing the concept and $V^c \subseteq \mathbf{V}$ is the attributes' domain: $V^c = \bigcup_{a \in A^c} V_a$.

Pair (A^c, V^c) is called the possible world or the structure of the concept c . Notice that within an ontology there may be two or more concepts with the same structure.

Definition 3 (Instance). *An instance of a concept c is described by the attributes from set A^c with values from set V_c . Thus, an instance of a concept c is defined as a pair:*

²<http://iwb.uidops.com/FedX>

$$\text{instance} = (id, v) \quad (3)$$

where id is a unique identifier of the instance in world (A, V) and v is the value of the instance, which is a tuple of type A^c and can be presented as a function:

$$v : A^c \rightarrow V^c \quad (4)$$

such that $v(a) \in V^a$ for $a \in A^c$. All instances of the same concept in an ontology are different with each other.

By $\text{Ins}(O, c)$ we denote the set of instances belonging to concept c in ontology O . We have

$$I = \bigcup_{c \in C} \text{Ins}(O, c) \quad (5)$$

In LOD, a description of a statement is represented by triples. A triple (s, p, o) where, s, p , and o stand for subject, predicate and object respectively. The subject of a triple is the URI identifying the described resource. The object can be the URI of another resource that is somehow related to the subject. The predicate indicates a specific relation exists between subject and object. A set of LOD triples is defined as a LOD graph.

Definition 4 (LOD Graph). *LOD graph is defined as a direct graph:*

$$G = (N, E) \quad (6)$$

where,

- $N = \{v_i | v_i \in I \cup C \cup V\}$: the set of vertices in G .
- $E = \{e(v_i, v_j) | v_i \in I \cup C, v_j \in N, e \in A \cup R \cup \sum\}$: the set of directed edges presented the predicates of all LOD triples. Each LOD triple (v_i, e, v_j) , where v_i , e , and v_j are the subject, predicate and object in the LOD triple, respectively.

4. A Technical Survey for LOD-federation Building

4.1. Federation Ontology Building

Definition 5 (Federation Ontology). *Given n component ontologies O_1, O_2, \dots, O_n corresponding to linked open data L_1, L_2, \dots, L_n , respectively. An federation ontology is an ontology:*

$$O^* = (C^*, \sum^*, R^*) \quad (7)$$

Where $\forall c_i \in \bigcup C_i, \exists c^* \in C^*, (c_i, c^*) \in M_c | c_i \equiv c^*$ and $\forall r_i \in \bigcup R_i, \exists r^* \in R^*, (r_i, r^*) \in M_r | r_i \equiv r^*$, M_c and M_r are conceptual map and relational map, respectively. We can say that the ontology O^* can replace all ontologies O_1, O_2, \dots, O_n .

To create federated ontology, the component ontologies should be integrated. Ontology integration is a well-known problem which is solved by various approaches, e.g. instance-base integration [13] and taxonomy-based integration [14,15]. However, the previous methods have not focused on specific LOD ontologies. The LOD ontology is not a specific-domain ontology. Linked Data sources are loosely coupled and typically controlled by independent third party providers which means that data schemata usually differ. Therefore, the existing efforts to develop application query across LODs primarily manually created schema level links between LOD ontologies. FactForge enables querying across LOD ontologies, and utilizes manually developed schema-level mappings of LOD ontologies to an upper level ontology called Proton [16]. ALOQUUS [17] used an upper ontology Proton matched to component LOD ontologies. These approaches lead to redundant and missing data, since many ontological entities either unused or mismatching into the component ontologies. Here, an LOD ontology integration is suggested to build the federation ontology.

Another system [18], which calls BLOOMS, is based on the idea of bootstrapping information on the LOD cloud. This approach is an utilization of the Wikipedia category hierarchy. Two ontologies, which are assumed to contain schema information, should be mapped by BLOOMS. These approach used additional knowledge, e.g. wikipedia, to enrich a concept semantic that yields to determine which concepts between two ontologies to align. However, this method did not consider contextual information, which leads to mismatching problem. An improved version, BLOOMS+, used a more sophisticated metric to determine which concepts between two ontologies to align, and considered contextual information to further support (or reject) an alignment.

We also proposed novel method for ontology integration [19] that effectively experimented with LOD ontologies. The main idea is underlining one that identifies the context of the concept by extending the concept's neighbor. Similarity between concepts is determined by similarity between their extended neighbors. For example, a concept *Java* in an ontology means a island, however, in another ontology, the concept *Java* means a programming language. If simply match between these two concepts, they are identified as similar concepts. However, an extended *Java*'s neighbor related to concepts such as *Programming*, *Eclipse*, *Java Sun*, *Java SE*, *JavaScript*, and *Java Applet*. Another extended *Java*'s neighbor regarding to *island*, *Philippians*, *Population*, and *Hindu-Buddhist*. According to these two extended neighbors, we can easily determine that the corresponding concepts are completely dissimilar.

- Longest Name Identify. Here a concept name can be considered as a noun phrase. Therefore, it is easy to identify the head noun of a concept name [20,21]. We generate a longest concept name containing the head noun that occurs on either WordNet, ODP, or Wikipedia.

- Local Neighbor Extraction. To extract a local neighbor of a concept, we only collect a set of children, parents, grandchildren, grandparents, uncle, and nephews of the concept belonging to its corresponding ontology.

- Global Neighbor Extraction. A global neighbor of a concept is derived from either WordNet, or ODP based on *ISA* relation (hierarchical relation), or from Wikimedia as way presented in [18]. The global neighbor reflects the domain-

independent concept for the corresponding concept, called domain-independent neighbor.

- Contextually expanded neighbor (CEN) generation. The global neighbor of a concept is matched to its local neighbor for the corresponding CEN generation. Here, we assume that the local neighbor of a concept can be used to identify specific domain for the concept in a context of the corresponding ontology. We align between the local neighbor and the global neighbor of the concept to identify a partial global neighbor in a context of the domain-specific ontology to which the concept belongs. We consider the partial global neighbor as the CEN of the concept. Therefore, each concept is represented by a CEN, which yields decisions which concepts are to be matched.

- Matching Phase. The matching phase is to calculate similarities between CENs to determine whether these corresponding concepts are matched or not. However, we do not exhaustively compute similarities between the concepts belonging to different ontologies, a method to skip the unmatchable pairs of concepts by propagating priorly matchable concepts is applied [14,15].

The aforementioned automatic approaches are to generate an integrated LOD ontology among LOD ontologies and schemas. However, these techniques are far from perfect for building an ontology federation by high requirement of an accurate matching, thus humans must be involved in the process. The automatic integrated LOD ontology is considered as an initial version for creating the ontology federation. This initial version should be manually refine and verify by experts. Recently, there are many mapping tools to help users to maintain mapping between ontologies like COMA++ [22], PROMPT [23], NeOn toolkit [24], and AlViz [25]. These tools usually consist of two mechanisms one that compare between ontologies to identify correspondences. Another one allows user to manually refine automatic mappings and supplement new mappings.

4.2. Query Decomposition

Linked Open Data is a successful version of Semantic Web. It provides data in a machine-readable form with links between related entities in the Web. Search in LOD not only supports keyword-based query, but also description query, a complex queries across different data sources. Therefore, the search in LOD is very similar to one in distributed/federated databases, but with less cooperation between the data sources, since LODs are typically controlled by independent third party which returns the results in a data format via a SPARQL protocol [8, 9]. The SPARQL protocol only defines how a query and the results are exchanged with the endpoints, therefore, how to select the relevant endpoint is an main issue in query decomposition.

We assume that LODs (datasets) L_1, L_2, \dots, L_n are supported by endpoints L_1, L_2, \dots, L_n , respectively. Each LOD L_i is considered as a LOD graph $G^i(N^i, E^i)$.

- User query creation: The federation ontology is provided for user to create user query. The query is depicted in either SPARQL or natural language by using concepts from federation ontology and query

- Query transformation: The user query is transformed into graph pattern $Q(N_q, E_q)$ [26]. Vertices of Q are comprising a set of variable vertices N_q^v and

constant vertices N_q^c , $N_q = N_q^v \cup N_q^c \subseteq N$. Edges of Q are formulae $e(v_i, v_j)$, with $v_i \in N_q^v$, $v_j \in N_q$. A result of Q on the graph is a mapping from vertices of Q to vertices of LOD graph G . The result

- Relevant dataset identification: Triple patterns of the graph query need to be evaluated only at relevant data sources that can assess results. In order to identify relevant datasets, we use maps M_c and M_r to identify entities in ontologies associated with LODs, which matched to entities in the graph pattern $Q(N_q, E_q)$. Several rules mentioned in [27] are then used to identify the relevant dataset (or endpoint), e.g., *if there is a dataset in which one of its vocabularies includes the predicate of a triple pattern, then it is relevant for the triple pattern.*

- Group triple patterns. To optimize query processing, the triple patterns being relevant to a single dataset should be executed together (as a conjunctive query) in a single subquery, instead of sequentially sending them at the respective endpoint. For a detailed formalization and technique, we refer the interested reader to [9], Schwarte et al. proposed mechanisms to group triple patterns (called exclusive group) that can be exclusively evaluated at single endpoints.

- Sub-query generation: A exclusive group for each dataset L_i is considered a conjunctive query or a subquery. A conjunctive query is denoted as sub-graph pattern $Q_i(N_q^i, E_q^i) \subset Q(N_q, E_q)$. This query $Q_i(N_q^i, E_q^i)$ is allocated to the corresponding endpoint E_i , and the result is denoted as a graph $R^i(N_r^i, E_r^i)$.

4.3. Result Integration

As aforementioned subsection, a query $Q(N_q, E_q)$ can be decomposed into sub-queries $Q_i(N_q^i, E_q^i)$ that may access the LODs L_i via respective endpoints S_i for results $R^i(N_r^i, E_r^i)$, $i=1..k$. Thus, the result integration can be considered as a graph integration problem, which is depicted as follows:

- Entity co-reference. According to Joshi et al. [17], the entity co-reference is defined as different resources referring to the same real world entity. To identify the entity co-reference, the equivalent properties such as *owl:sameAs* or *skos:exactMatch* are useful, two entities are co-reference if they either are tied by an equivalent property or have common well-known equivalent properties.

- Result integration. Using entity co-reference technique to explicitly identify correspondences between partial subquery results. The correspondences are to connect the partial results to reconstruct the final result. The detail techniques such as bound joins and semijoin were introduced in [10] and [8], respectively.

5. Illustrated Example

We consider a situation that a user want to know information of an actor whose name is Martin P.Robinson, including date of birth, performance, place of birth and its population.

- User query creation: The query can be formulated in natural language as follows: Identify the date of birth, place of birth and its population, and performances of an actor whose name is Martin P.Robinson.

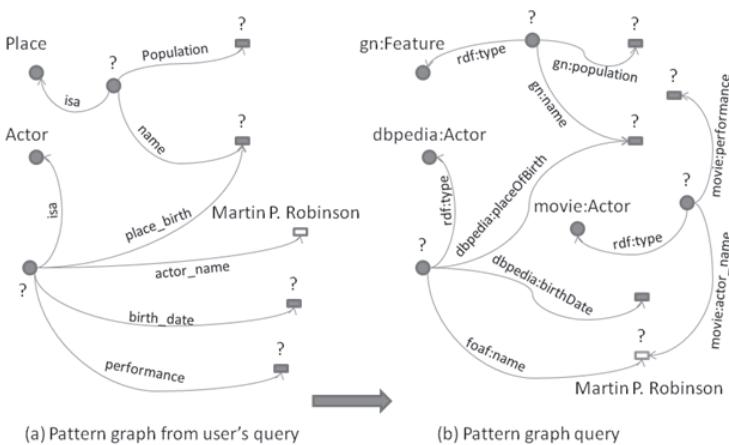


Figure 1. User's Query Transformation

- Query transformation: The query is transformed to a pattern graph (see Figure 1 (a)).
- Relevant dataset identification: This graph matched to the federated ontology to identify relevant dataset including Sparql-endpoints <http://dbpedia.org/sparql>, <http://data.linkedmdb.org/sparql> and <http://geosparql.org/>.
- Group triple patterns: The pattern graph query ((see Figure 1 (b)) is generated by grouping triple pattern being relevant to a single dataset, which should be executed together.
- Sub-query generation: This graph query is decompose into three subsequent queries as follows:

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?birth_date ?birth_place
WHERE {
?actor a dbpedia:Actor ; foaf:name "Martin_P._Robinson"@en; dbpedia:birthDate ?birth_date; dbpedia:birthPlace ?birth_place
}
```

Its SPARQL-endpoint is <http://dbpedia.org/sparql>

```
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>
```

```
SELECT ?actor_name ?performance_film
```

```
WHERE {
```

```
?film movie:actor ?actor.
```

```
?actor movie:actor_name "Martin_P._Robinson"@en.
```

```
?actor movie:performance ?performance.
```

```
?performance movie:performance_film ?performance_film
```

```
}
```

Its SPARQL-endpoint is <http://data.linkedmdb.org/sparql>

```
SELECT ?name ?pop
```

```
WHERE {
```

```
?link gs:within(40.157623 -74.855347 41.077281 -73.586426) .
?link gn:name ?name.
?link gn:population ?pop.
}
```

Its SPARQL-endpoint is <http://geosparql.org/>

- Result Integration: The partial results can be integrated using the entity co-reference such as owl:sameAs and skos:exactMatch or entity similarity measures.

6. Conclusions

In this study, a technical survey of recent works supporting for LOD-federation building was presented. LOD-federation is considered as an abstract level above endpoints, which includes a global ontology and mechanisms for query processing. An effective method of ontology integration was suggested to build the federated ontology [19]. This method was evaluated as a significant work in comparison with previous works such as BLOOMS [18] for LOD ontology matching and other methods such as S-Match [28] and AROMA [29]. The mechanisms for query processing include query decomposition and result integration. The query decomposition is to translate a query to a LOD-federation into subsequent LODs. Graph pattern [26] and relevant rules [27] are used to identify relevant datasets. To optimize query processing, the triple patterns, which is relevant to a single dataset, should be executed together (as a conjunctive query) in a single subquery, instead of sequentially sending them at the respective endpoint [9]. A result integration method, using co-reference entity in [17] and join operations in [10,8], was introduced. In the future work, a conflict resolution will be considered to solve the inconsistency in integrated result.

References

- [1] Berners-Lee T., Hendler J., Lassila O.: The Semantic Web, *Scientific American*, 5, 284(2001), 35 - 43.
- [2] Haase P., Martha T., Ziller M.: An Evaluation of Approaches to Federated Query Processing over Linked Data. In Proceedings of the 6th International Conference on Semantic Systems, ACM New York, NY, USA (2010), 1-9.
- [3] Berger S., Schreifl M.: From Federated Databases to a Federated Data Warehouse System. Proceedings of the 41st Hawaii International Conference on System Sciences, IEEE Computer Society Washington, DC, USA (2008), 394.
- [4] Heimbigner, D., McLeod, D.: A Federated Architecture for Information Management. *ACM Transactions on Information Systems* 3(3),(1985), 253-278.
- [5] Sheth, A., Larson, J.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys* 22(3), (1990), 183-236.
- [6] Kossmann, D.: The State of the Art in Distributed Query Processing. *ACM Computing Surveys* 32(4),(2000), 422-469.
- [7] Josifovski, V., Schwarz, P., Haas, L., Lin, E.: Garlic: A New Flavor of Federated Query Processing for DB2. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin (2002), 524-532.

- [8] Gorlitz O., Staab S.: Federated Data Management and Query Optimization for Linked Open Data. In New Directions in Web Data Management 1, Studies in Computational Intelligence Volume 331 (2011), 109-137.
- [9] Schwarte A., Haase P., Hose K., Schenkel R., Schmidt M.: FedX: Optimization Techniques for Federated Query Processing on Linked Data. International Semantic Web Conference (2011), 601-616.
- [10] Schwarte, A.: FedX: Optimization Techniques for Federated Query Processing on Linked Data. Master's thesis, Saarland University, Germany, 2011.
- [11] Nguyen N.T. (2008): Inconsistency of Knowledge and Collective Intelligence. *Cybernetics and Systems* 39(6), 542-562.
- [12] Nguyen N.T. (2007): A Method for Ontology Conflict Resolution and Integration on Relation Level. *Cybernetics and Systems* 38(8).
- [13] Doan, A. H., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: a machine learning approach. *Handbook on Ontologies in Information Systems*, Springer Verlag (2003), 397-416.
- [14] Duong T.H., Nguyen N.T., Jo G.S.: A Hybrid Method for Integrating Multiple Ontologies. *Cybernetics and Systems* 40, 2 (2009), 123-145.
- [15] Duong T.H., Jo G.S.: Enhancing performance and accuracy of ontology integration by propagating priorly matchable concepts, *Neurocomputing*, 88(2012), 3-12.
- [16] Damova, M., Kiryakov, A., Simov, K., Petrov, S.: Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I., eds.: In Proceedings of the Fifth International Workshop on Ontology Matching. CEUR Workshop Proceedings (November 2010)
- [17] Joshi A. K., Jain P., Hitzler P., Yeh P. Z., Verna K., Sheth A. P., and Damova M., Alignment-based Querying of Linked Open Data, in Proceedings of Ontologies, DataBases, and Applications of Semantics (ODBASE) 2012, Rome, Italy.
- [18] Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In International Semantic Web Conference (2010), 402-417.
- [19] Duong T.H., Truong H.B., Nguyen N.T.: Local Neighbor Enrichment for Ontology Integration, ACIIDS 2012, Lecture Notes in Computer Science(2012),156-166.
- [20] Duong T.H., Nguyen N.T., Jo G.S.: A Method for Integration of WordNet-based Ontologies Using Distance Measures. In KES 2008. LNAI 5177(2008), 210-219.
- [21] Duong T.H., Nguyen N.T., Jo G.S.: A Method for Integration across Text Corpus and WordNet-based Ontologies. In IEEE/ACM/WI/IAT 2008 Workshops Proceedings, IEEE Computer Society (2008), 1-4.
- [22] Do H.H. Schema matching and mapping-based data integration. PhD thesis, Department of Computer Science, Universitat Leipzig (2006).
- [23] Noy N.F., Musen M.A.: The PROMPT suite: Interactive tools for ontology merging and mapping. *Int J Hum Comput Stud* 59(6),(2003), 983-1024.
- [24] Le D.: Matching ontologies for context: The neon alignment plug-in. Tech. Rep. Deliverable 3.3.2, IST NeOn IP, NeOn. <ftp://ftp.inrialpes.fr/pub/exmo/reports/neon-332.pdf> (2008)
- [25] Lanzenberger M., Sampson J. Alviz: a tool for visual ontology alignment. In: Proceedings of the conference on information visualization (IV), London, July 2006. IEEE Computer Society, Washington, DC, 430-440.
- [26] Wagner, A., Ladwig, G., Tran, T.:Browsing-oriented Semantic Faceted Search, DEXA'11 Proceedings of the 22nd international conference on Database and expert systems applications - Volume Part I, Springer-Verlag Berlin, Heidelberg, (2011), 303-319.
- [27] Akar Z., Halac T. G., Ekinici E. E., Dikenelli O.: Querying the Web of Interlinked Datasets using VOID Descriptions, In Workshop: Linked Data on the Web (2012).
- [28] Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an Algorithm and an Implementation of Semantic Matching. In ESWS (2004), LNCS 3053, 61-75.
- [29] David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In CIKM 06: The 15th ACM International Conference on Information and Knowledge Management, New York, NY, USA, ACM (2006), 830-831.

A Deduction-based System for Formal Verification of Agent-ready Web Services

Radosław KLIMEK¹

AGH University of Science and Technology, Krakow, Poland

Abstract. The work concerns the formal verification of systems using logical inference and considers web service models expressed in BPEL. Agent technologies and web services may be consistent with some features and mutual goals like independence, interoperability, service and aims discovery, etc. A method for an automatic extraction of logical specifications, considered as a set of temporal logic formulas, for web services is proposed. The generation procedure is based on predefined BPEL-based patterns. The patterns are considered primitives, which enables the transformation to temporal logic formulas. The inference process is based on the semantic tableaux method. The proposed method of formal verification could be helpful in avoiding arbitrariness and laxity with agent-oriented web services.

Keywords. verification, deduction, web services, BPEL, agents, design patterns, temporal logic, semantic tableaux

1. Introduction

Agent technologies, including systems implemented by web services, are developed dynamically. *Intelligent agents* are software entities composed of three important features: autonomy, reactivity, and communication ability. A recent trend has been to introduce, combine and integrate agents and web service technologies, c.f. [1,2,3]. The agents' distributed structure seems to be particularly suitable for complex and BPEL-based web service compositions. The BPEL language is the primary candidate for the dynamic execution of composing existing web services and is a crucial component of business-to-business infrastructures. The BPEL-based services could be modeled using BPMN notations. The obtained software models require careful verification using mature tools to ensure that the received software products are reliable. The work focuses on formal approach of the BPEL-based services and the objective is to get intuitive methods and tools which enable the formal verification of developed systems using deductive reasoning.

BPEL (Business Process Execution Language) is an executable XML-based language that models of interactions between web services. BPEL is relatively simple, human-readable and facilitates the cooperation between business people and developers. It belongs to SOA (Service-oriented Architecture) which provides a set of principles for web-based applications, and was standardized by OASIS [4]. BPEL is associated with BPMN (Business Process Management Notation) designed for modeling and planning

¹Corresponding Author: Radosław Klimek, AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Krakow, Poland; E-mail: rklimek@agh.edu.pl

business processes. BPMN is easier to understand for business people not familiar with programming. BPMN is not executable whereas BPEL is, though translating from one to the other remains a challenge.

The work concerns the deductive reasoning used for formal verification. The need to build a system's logical specifications considered as a set of temporal logic formulas $\{p_1, \dots, p_n\}$ is a difficult issue. When n is large, which is not uncommon, in practice it is not possible to build a logical specification manually, hence the need to automate this process. Automation enables bridging the gap between the naturalness of the deductive inference and the difficulty of its practical application. The lack of automation solutions is an obstacle to the practical use of inference for formal verification. The inference process is based on the semantic tableaux method which has many advantages to compare with other deduction strategies. Deduction-based formal verification is essential for sustainable verification leverage, characterized by intuitiveness, a top-down way of thinking, logic-based reasoning, coverage of infinite computations, etc.

2. Motivation, Contribution and Related Works

The work's general motivation is the lack of satisfactory results of the practical use of deduction-based formal verification. BPEL-based web service models, whose properties could be expressed in temporal logic, for agent systems seem to be an important class of systems suitable for formal verification. The main motivation is the lack of tools for automatic extraction of logical specifications, i.e. a set of temporal logic formulas.

The key contribution is the automation of the process of logical specification generation. Theoretical possibilities of such automation are discussed. The generation algorithm for selected BPEL design patterns is presented. Another contribution is the use of a non-standard method for deduction which is the semantic tableaux method for temporal logic. The main contribution is a complete deduction-based system, including its architecture, which enables automated and formal verification of BPEL models/programs.

Let us discuss some current research directions. Work by Dury et al. [5] discusses business workflows for formal verification using model checking. In work by Brambilla et al. [6] some aspects of workflows and temporal logic are considered but formulas are created manually and formal verification is not widely discussed. Duan et al. [7] introduced a logic model to specify the semantics of workflow and make a deduction of the weakest precondition for workflow. Another important research direction is the verification of business processes using Petri nets [8]. A different direction of formal verification is related to π -calculus, such as the work by Ma et al. [9], designed for business processes and the BPEL environment. Work by Dong et al. [10] discusses formal models for BPEL with the aim of model checking. Rule-based systems also enable verification of system formal properties, c.f. work by Wojnicki [11]. Mobile robots navigation, c.f. work by Turek et al. [12], seems to be an important area of application for agents. Last but not least is work by Morimoto [13] and an interesting survey of formal verification for business processes. It discusses automata, model checking, communicating sequential processes, Petri nets, and Markov networks. All these issues are discussed in the context of business process management and web services. However all of the research themes mentioned above are different from the approach presented in the work. A very preliminary version of this is [14]. Work [15] discusses BPMN models for formal verification using a deductive approach.

3. Logical Background

Temporal Logic TL is a valuable formalism, e.g. [16,17], which has strong application in the area of software engineering for the specification and verification of models. It exists in many varieties, though considerations in this paper are limited to the *Linear Temporal Logic* LTL, i.e. logic with a linear time structure. This logic provides specification of both *liveness* properties, which ensure that desirable states are obtained in the course of the computation, and *safety* properties, which ensure that undesirable states are never obtained.

It is obvious that temporal logic is always considered in the context of a time structure. Accepted properties of a time structure provide a background for every temporal logic. Thus, its properties are crucial in relation to valid formulas and theorems of some logic. Considerations in this paper are limited to the *smallest*, or *minimal*, *temporal logic* (e.g. [18]), also known as class K temporal logic. Thus, the following formulas may be considered as significant examples of the minimal temporal logic: $\text{action} \Rightarrow \diamond \text{reaction}$, $\square(\text{request} \Rightarrow \diamond \text{grant})$, $\diamond \text{liv}$, $\square \neg(\text{bad})$ or $\square \neg(\text{event1} \wedge (\text{event2} \vee \text{event3}))$, etc. The syntax of temporal logic can be considered as an extension of classical propositional logic, c.f. [16], and is not discussed here.

The work focuses on formal deduction as a method of verification in opposition to the states' exploration methods. Inference based on traditional methods, i.e. axiomatization and resolution, is of course possible but a much more interesting alternative seems to be the *semantic tableaux*, or the *truth tree* method. The method is well known in classical logic, can be applied in modal logic [19] and is a decision procedure for checking the ability to satisfy a formula. It represents reasoning by contradiction. It has some advantages with respect to the traditional axiomatic approach. The method is based on formula decomposition. At each step of the well-defined procedure, formulas become simpler as logical connectives are removed. At the end of the decomposition procedure, all branches of the received tree are searched for contradictions. When any branch of the truth tree has a contradiction, it means that the branch is *closed*. When all branches are closed, then the tree is closed. In the classical reasoning approach, starting from axioms, longer and more complicated formulas are generated and derived. Formulas growing longer and longer with every step, and only one of them will lead to the verified formula. The semantic tableaux method is characterized by the reverse strategy. Though we start with a long and complex formula, it becomes less complex and shortens with every step of the decomposition procedure. An important advantage is that the method provides, through so-called *open* branches of the semantic tree, information about the found's error's source. Another advantage, particularly with regard to the resolution approach, is the fact that tableaux are global, goal-oriented and "backward", while resolution is local and "forward". Tableaux also seem to be more machine-oriented.

4. Pattern Oriented Design

In work [20], Riehle and Zullighoven described *patterns* as: "the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts". Design patterns are significant for the approach introduced in this work as they enable the automation of the logical specifications generation process. They constitute a kind of primitives which en-

able the mapping of design patterns to logical specifications. Thus, BPEL-based design patterns are introduced. The proposed method of the automatic extraction of logical specifications is based on the assumption that the entire BPEL program is built using only predefined design patterns. In fact, this assumption cannot be recognized as a restriction since it enables receiving correct and well-composed systems.

Some general notions must be defined first. Every design pattern is associated with temporal logical formulas describing the pattern's properties.

Definition 1 An elementary set of formulas over atomic formulas $a_{i,i=1,\dots,n}$ is denoted $\text{pat}(a_i)$, or simply $\text{pat}()$, as a set of temporal logic formulas $\{f_1, \dots, f_m\}$ such that all formulas are well-formed.

The proposed temporal logic formulas should describe both safety and liveness properties for each pattern. $\text{Pat1}(a,b) = \{a \Rightarrow \diamond b, \square \neg(a \wedge b)\}$ and $\text{Pat2}(a,b,c) = \{a \Rightarrow \diamond b \wedge \diamond c, \square \neg(a \wedge (b \vee c))\}$ are examples of elementary sets.

The whole BPEL model can be quite complex, including nesting patterns, which why there is a need to define a symbolic notation which enables the representation of a potentially complex structure.

Definition 2 The logical expression W_L is a structure created using the following rules:

- every elementary set $\text{pat}(a_i)$, where $i > 0$ and every a_i is an atomic formula, is a logical expression,
- every $\text{pat}(A_i)$, where $i > 0$ and every A_i is either
 - * an atomic formula a_j , where $j > 0$, or
 - * a set $\text{pat}(a_j)$, where $j > 0$ and a_j is an atomic formula, or
 - * a logical expression $\text{pat}(A_j)$, where $j > 0$

is also a logical expression.

The example of logical expression is $\text{Seq}(a, \text{Seq}(\text{Flow}(b, c, d), \text{Switch}(e, f, g)))$ which is intuitive, in that it shows the sequence that leads to the sequence of a parallel split (flow) and then conditional execution (switch) of some activities.

Now, BPEL programs/models and their design patterns are discussed. BPEL supports structured programming and includes both sequence, i.e. executing commands in order, and flow, i.e. executing commands in concurrency. It also includes conditional executions and loops (while). From the viewpoint of the approach presented in this work it is important to introduce some restrictions on BPEL programs. This results from the introduction design patterns for BPEL programs. On the other hand, patterns always lead to obtaining a structural form of the BPEL program and as it is known the language supports structured programming. The following design patterns are introduced: *sequence*, *flow*, *switch* and *while-loop* for iteration. These patterns are based on structured BPEL activities, i.e. they describe processes by structuring basic activities. This pattern set can be regarded as a minimal set, however it is sufficient to define all the basic situations when developing web services. Other types of patterns and activities are omitted to focus on the overview of the method, i.e. extraction of the logical specification in order to use deductive reasoning for the formal verification of BPEL programs.

The *sequence pattern* indicates activities executed in a predefined sequential order using sequence activities.

```
<sequence>
    activity A1
    activity A2
</sequence>
```

The execution order follows the specifications of activities placed between tags. The *flow pattern* indicates that after activity execution, control flow is forked and then two activities are executed concurrently.

```
<sequence>
    activity A1
    <flow>
        activity A2
        activity A3
    </flow>
</sequence>
```

The flow pattern is finished when both forked activities are done. The *switch pattern* indicates the conditional behavior.

```
<switch>
    <case condition="bool-expr">   <!-- activity A1 -->
        activity A2
    </case>
    <otherwise>
        activity A3
    </otherwise>
</switch>
```

First, the condition which constitutes an activity is evaluated. When the condition is true, then the activity specified with the “case” tag is executed else the activity specified with the “otherwise tag” is executed. The “otherwise” tag is mandatory. The *while pattern* indicates the possibility of an iterative way of execution for a certain activity.

```
<while condition="bool-expr">   <!-- activity A1 -->
    activity A2
</while>
```

First, the loop condition which constitutes an activity is evaluated. The loop activity is executed as long as the boolean expression in the condition activity is true. The while pattern is finished when the loop activity is completed except when the loop condition has never been fulfilled.

Special cases are basic activities handling message flows. Let us focus our attention on the invoke and receive activities:

```
<invoke> <receive>
```

An invoke activity enables calling an other web service that has been defined as a partner. A receive activity specifies a partner that can be invoked. “Partner” is a required attribute of these activities which lead us to believe that messages can be exchanged with the outside world (pool).

Definition 3 *The message flow MesFlo is a set of edges MesFlo = {(v, w) : v, w ∈ A ∧ v = w}, where A is a set of all activities used in the BPEL program.*

This saves all message flows. Every BPEL program may be considered as a directed graph, where activities are vertices and control flows are edges, c.f. Figure 2. There are two basic types of messages, or message flows, i.e. *Invoke* and *Receive* the meaning of which is understandable. Other types of messages are omitted to simplify the consideration.

5. Logical Specification Modeling

Logical properties of all design patterns are expressed in temporal logic formulas and stored in the predefined and fixed *logical properties set P*. These formulas are used in the algorithm for generating logical specifications. An example of such a predefined set *P* (plain ascii text) is shown in Figure 1. Most elements of the *P* set, i.e. two temporal logic operators, classical logic operators, etc. are not in doubt. a_1, a_2 and a_3 are atomic formulas and constitute formal arguments for a pattern. $\diamond a$ means that at sometime (or eventually in the future) activity a is active, i.e. the token reached the activity, and this fact has been established when the signal edge (the falling or the rising edge) is transited. $c(a)$ means that the logical condition associated with the activity a has been evaluated and is satisfied. This logical condition is also satisfied when the falling edge is transited. All formulas describe both safety and liveness properties for every pattern. On the occasion of the above considerations, let us introduce the following two notations. $i(a)$ means that the invoke message for the activity a is called and $i(a)$ is also satisfied when the falling edge of the originate activity is transited. $r(a)$ means that the receive message for the activity a is called and $r(a)$ is also satisfied when the falling edge of the originate activity is transited. Suppose also that the invoke message is always accompanied by the other side of the receive message. The last simplified assumption is that there is at most one invoke and one receive message for every activity.

```

Sequence(a1,a2):      /* ver. 5.12.2012
a1 => <>a2
[] ~(a1 & a2)
Flow(a1,a2,a3):
a1 => <>a2 & <>a3
[] ~(a1 & (a2|a3))
Switch(a1,a2,a3):
a1 & c(a1) => <>a2
a1 & ~c(a2) => <>a3
[] ~((a1 & a2) | (a1 & a3) | (a2 & a3))
Loop-While(a1,a2):
a1 & c(a1) => <> a2
a1 & ~c(a1) => ~<> a2 [] ~(a1 & a2)

```

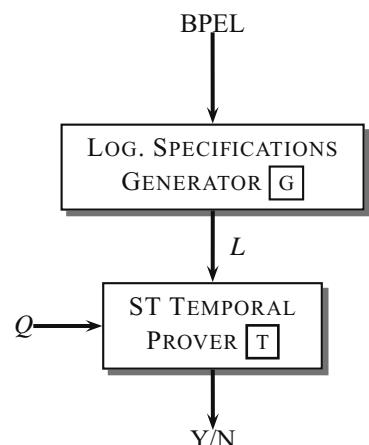


Figure 1. Predefined set of pattern temporal properties (left) and deduction-based verification system (right)

A logical specification which is generated from a logical expression is defined.

Definition 4 A logical specification L consists of all formulas obtained from a logical expression using the algorithm Π , i.e. $L(W_L) = \{f_i : i > 0 \wedge f_i \in \Pi(W_L, P)\}$, where f_i is a TL formula.

However, generating a logical specification is not a simple summation of formula collections resulting from a logical expression. The sketch of the generation algorithm is presented below. The generation process has two inputs. The first one is a logical expression W_L which is a kind of variable, i.e. it varies for every BPEL model. The second is a predefined set *P*, a kind of constant in that it contains predefined temporal properties for every BPEL pattern.

The output of the generation algorithm Π is a logical specification understood as a set of temporal logic formulas. The generation algorithm sketch is given below.

1. At the beginning, the logical specification is empty, i.e. $L := \emptyset$;
2. Patterns are processed from the most nested pattern to be located more towards the outside and from left to right;
3. If the currently analyzed pattern consists only of atomic formulas, and none of the atomic formulas belong to the set $Invoke \cup Receive$, the logical specification is extended, by summing sets, by formulas linked to the type of pattern analyzed, i.e. $L := L \cup pat()$;
4. If any argument is a pattern itself, then the logical disjunction of all its arguments, including nested arguments, is substituted in place of the pattern;
5. If any atomic formula $a = v_1$, where (v_1, w_1) belongs to the set $Invoke$, then the logical conjunction of a and $i(w_1)$, i.e. $a \wedge i(w_1)$, is substituted in place of an activity a ; if any atomic formula $a = w_2$, where (v_2, w_2) belongs to the set $Receive$, then the logical conjunction of a and $r(v_2)$, i.e. $a \wedge r(v_2)$, is substituted in place of an activity a ;
6. For every (v, w) that belongs to the set message flows $Invoke \cup Receive$ do $L := L \cup \{v \Rightarrow \Diamond w\}$.

The above algorithm extends similar one in work [15]. All patterns of the logical expression are processed one by one and the algorithm always halts. All parentheses are paired. Let us supplement the algorithm by some examples. *Seq* is an alias of *Sequence*. The example for step 3: $Seq(a, b)$, gives $L = \{a \Rightarrow \Diamond b, \Box \neg(a \wedge b)\}$ and $Switch(a, b, c)$ gives $L = \{a \wedge c(a) \Rightarrow \Diamond b, a \wedge \neg c(a) \Rightarrow \Diamond c, \Box \neg((a \wedge b) \vee (a \wedge c) \vee (b \wedge c))\}$. The example for step 4: $Flow(Seq(a, b), c, d)$ leads to $L = \{a \Rightarrow \Diamond b, \Box \neg(a \wedge b)\} \cup \{(a \vee b) \Rightarrow \Diamond c \wedge \Diamond d, \Box \neg((a \vee b) \wedge (c \vee d))\}$. Steps 5 and 6 are illustrated during the analysis of the diagram from Figure 2.

6. A Deduction System

The outline architecture of the proposed deduction-based system using the semantic tableaux method for BPEL design patterns is presented in Figure 1. The first component G generates logical specifications. The generation of formulas is performed automatically by extracting directly from the design patterns contained in a BPEL program. A specification is treated as a conjunction of formulas $p_1 \wedge \dots \wedge p_n = P$ and every p_i is a specification formula generated during the extraction. These formulas constitute a logical specification L . The Q formula is a desired property of a BPEL program (system). The formula is expressed in temporal logic and identified by the analyst describing the expected properties of the investigated system. Both the specification of a system and the examined properties constitute an input to the T component, i.e. *Semantic Tableaux Temporal Prover*, or shortly *ST Temporal Prover*, which enables the automated reasoning. The input for this component is the formula $P \Rightarrow Q$, or, more precisely:

$$p_1 \wedge \dots \wedge p_n \Rightarrow Q \quad (1)$$

After negation of formula (1), it is placed at the root of the inference tree and decomposed using the semantic tableaux method's well-defined rules. If the inference tree is closed,

this means that the initial formula (1) is true. Broadly speaking, the output of the \boxed{T} component, and therefore also the output of the whole deductive system, is the answer Yes/No in response to any introduction of a new tested property for the BPEL model.

The whole verification procedure can be summarized thus:

1. Automatic generation of a logical specification (the \boxed{G} component);
2. Introduction a property Q of the system;
3. The automatic inference using semantic tableaux (the \boxed{T} component) for the whole formula (1).

Steps 1 to 3, taken as a whole or individually, may be processed many times, whenever the specification of the BPEL program is changed (step 1) or there is a need for a new inference due to the revised system's specification (steps 2 or 3).

7. Illustration of the Approach

Let us consider a simple example to illustrate the approach presented in the work. Figure 2 shows a diagram of a BPEL program. There are two regular expressions $Pool1 = Seq(Seq(Flow(a,b,c), Switch(d,e,f)), Seq(g,h))$ and $Pool2 = Seq(i,j)$. Message flow sets are $Invoke = \{(g,i)\}$ and $Receive\{(j,h)\}$. A single liveness property expressed as a formula (3) is to be verified but it is also possible to analyze other properties, including safety. Logical specification L is built in the following steps. At the beginning the specification of a model is $L = \emptyset$. The most nested patterns are *Flow* and *Switch*. *Flow* gives $L = L \cup \{a \Rightarrow \diamond b \wedge \diamond c, \square \neg(a \wedge (b \vee c))\}$ and *Switch* gives $L = L \cup \{d \wedge c(d) \Rightarrow \diamond e, d \wedge \neg c(d) \Rightarrow \diamond f, \square \neg((d \wedge e) \vee (d \wedge f) \vee (e \wedge f))\}$. The next considered pattern is nested *Sequence* for which an arguments' disjunctions are considered. It gives $L = L \cup \{(a \vee b \vee c) \Rightarrow \diamond(d \vee e \vee f), \square \neg((a \vee b \vee c) \wedge (d \vee e \vee f))\}$. Then the message flows are included that leads to $L = L \cup \{(g \wedge i(i)) \Rightarrow \diamond(h \wedge r(j)), \square \neg((g \wedge i(i)) \wedge (h \wedge r(j)))\}$. The outermost *Sequence* gives $L = L \cup \{(a \vee b \vee c \vee d \vee e \vee f) \Rightarrow \diamond((g \wedge i(i)) \vee (h \wedge r(j))), \square \neg((a \vee b \vee c \vee d \vee e \vee f) \wedge ((g \wedge i(i)) \vee (h \wedge r(j))))\}$. The second pool gives $L = L \cup \{i \Rightarrow \diamond j, \square \neg(i \wedge j)\}$. The last step includes message flows

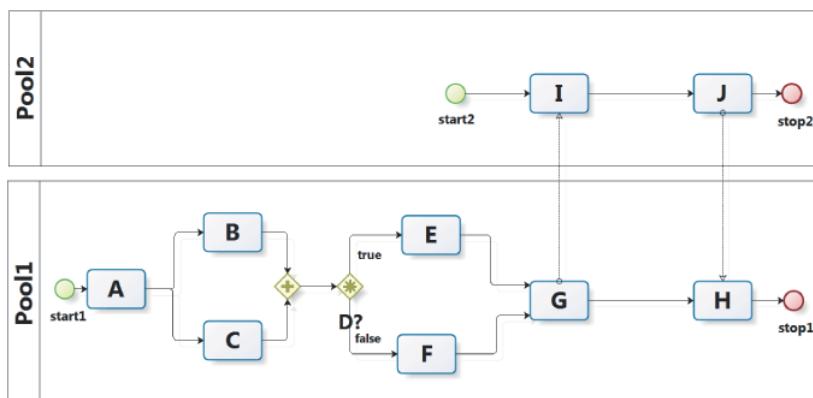


Figure 2. Visualization of a web service model expressed in BPEL

$L = L \cup \{g \Rightarrow \diamond i, j \Rightarrow \diamond h\}$. Thus, the resulting specification contains all above mentioned formulas

$$L = \{a \Rightarrow \diamond b \wedge \diamond c, \square \neg(a \wedge (b \vee c)), \dots, g \Rightarrow \diamond i, j \Rightarrow \diamond h\} \quad (2)$$

The examined property can be

$$a \Rightarrow \diamond h \quad (3)$$

which means that if a is chosen then sometime in the future h is completed. The whole formula to be analyzed using the semantic tableaux method is

$$\begin{aligned} & (a \Rightarrow \diamond b \wedge \diamond c) \wedge (\square \neg(a \wedge (b \vee c))) \wedge \\ & (d \wedge c(d) \Rightarrow \diamond e) \wedge (d \wedge \neg c(d) \Rightarrow \diamond f) \wedge \\ & (\square \neg((d \wedge e) \vee (d \wedge f) \vee (e \wedge f))) \wedge ((a \vee b \vee c) \Rightarrow \diamond(d \vee e \vee f)) \wedge \\ & (\square \neg((a \vee b \vee c) \wedge (d \vee e \vee f))) \wedge \\ & ((g \wedge i(i)) \Rightarrow \diamond(h \wedge r(j))) \wedge (\square \neg((g \wedge i(i)) \wedge (h \wedge r(j)))) \wedge \\ & ((a \vee b \vee c \vee d \vee e \vee f) \Rightarrow \diamond((g \wedge i(i)) \vee (h \wedge r(j)))) \wedge \\ & (\square \neg((a \vee b \vee c \vee d \vee e \vee f) \wedge ((g \wedge i(i)) \vee (h \wedge r(j))))) \wedge \\ & (i \Rightarrow \diamond j) \wedge (\square \neg(i \wedge j)) \wedge (g \Rightarrow \diamond i) \wedge (j \Rightarrow \diamond h) \Rightarrow (a \Rightarrow \diamond h) \end{aligned} \quad (4)$$

Formula (2) represents the output of the \boxed{G} component in Figure 1. Formula (4) provides a combined input for the \boxed{T} component in Figure 1. Presentation of a full reasoning tree for formula (4) exceeds the size of the work. The tree contains many hundreds of nodes. The formula is true and the examined property (3) is satisfied in the considered model.

8. Conclusion

The work proposes a method for the automatic generation of logical specifications from software models based on design patterns. It enables formal verification using temporal logic and semantic tableaux provers of BPEL-oriented web service models. The work presents an algorithm for generating logical specifications of models expressed in BPEL. The method's advantage is that it provides an innovative concept for process verification which might be done for any model created using the BPEL language and predefined patterns. Future research should extend the results to other BPEL patterns including sub-tasks. The next research step may also involve a detailed case study and enriched logics.

Acknowledgements

This work has been co-financed by the European Union, Human Capital Operational Programme, SPIN project no. 502.120.2066/C96 and co-financed by the AGH Research Fund no. 11.11 120.859.

References

- [1] T. R. Payne, Web services from an agent perspective, *IEEE Intelligent Systems*, vol. 23, pp. 12–14, 2008.
- [2] J. Lee, S.-J. Lee, H.-M. Chen, and C.-L. Wu, Composing web services enacted by autonomous agents through agent-centric contract net protocol, *Information & Software Technology*, vol. 54, no. 9, pp. 951–967, 2012.
- [3] Y. Li, W. Shen, and H. Ghenniwa, Agent-based web services framework and development environment, *Computational Intelligence*, vol. 20, no. 4, pp. 678–692, 2004.
- [4] OASIS, Web services business process execution language version 2.0, tech. rep., OASIS, <http://docs.oasis-open.org/wsdl/v2.0/wsdl-v2.0.pdf>, 2007.
- [5] A. Dury, S. Boroday, A. Petrenko, and V. Lotz, Formal verification of business workflows and role based access control systems, *Proc. of 1st Int. Conf. on Emerging Security Information, Systems and Technologies (SecurWare 2007), October 14–20, 2007, Valencia, Spain* (L. Peñalver and et al., eds.), pp. 201–210, IEEE Computer Society, 2007.
- [6] M. Brambilla, A. Deutsch, L. Sui, and V. Vianu, The role of visual tools in a web application design and verification framework: A visual notation for Itl formulae, *Proc. of 5th Int. Conf. on Web Engineering (ICWE 2005), July 27–29, 2005, Sydney, Australia* (D. Lowe and M. Gaedke, eds.), vol. 3579 of LNCS, pp. 557–568, Springer, 2005.
- [7] Z. Duan, A. J. Bernstein, P. M. Lewis, and S. Lu, Semantics based verification and synthesis of bpel4ws abstract processes, *Proc. of the IEEE Int. Conf. on Web Services (ICWS'04), June 6–9, 2004, San Diego, California, USA*, pp. 734–737, IEEE Computer Society, 2004.
- [8] W. M. P. van der Aalst, Making work flow: On the application of petri nets to business process management, *ICATPN*, vol. 2360 of LNCS, pp. 1–12, Springer, 2002.
- [9] S. Ma, L. Zhang, and J. He, Towards formalization and verification of unified business process model based on pi calculus, *Proc. ACIS International Conference on Software Engineering Research, Management and Applications*, pp. 93–101, 2008.
- [10] R. Dong, Z. Wei, and X. Luo, Model checking behavioral specification of bpel web services, *Proceedings of the World Congress on Engineering (WCE 2008), July 2–4, 2008, London, U.K.*, vol. I, pp. 198–203, Newswood Limited, 2008.
- [11] I. Wojnicki, Implementing general purpose applications with the rule-based approach, *RuleML'2011: Proceedings of the 5th international conference on Rule-based reasoning, programming, and applications*, (Berlin, Heidelberg), pp. 360–367, Springer, 2011.
- [12] W. Turek, R. Marcjan, and K. Cetnarowicz, Agent-based mobile robots navigation framework, *Proceeding of the 6th International Conference on Computational Science (ICCS), Reading, UK, May 28–31, 2006, Part III*, vol. 3993 of LNCS, pp. 775–782, Springer, 2006.
- [13] S. Morimoto, A survey of formal verification for business process modeling, *Proc. of 8th Int. Conf. Computational Science (ICCS 2008), June 23–25, 2008, Kraków, Poland, Part II* (M. Bubak and et al., eds.), vol. 5102 of LNCS, pp. 514–522, Springer, 2008.
- [14] R. Klimek and G. Rogus, Correctness formal analysis of web services [in polish], *Proc. of 13th National Conf. of Software Engineering (KKIO 2011), 12–15 September 2011, Czarna, Bieszczady, Poland* (J. Górski and C. Orłowski, eds.), pp. 213–220, PWNT, 2011.
- [15] R. Klimek, Towards formal and deduction-based analysis of business models for soa processes, *Proceedings of 4th International Conference on Agents and Artificial Intelligence (ICAART 2012), 6–8 February, 2012, Vilamoura, Algarve, Portugal* (J. Filipe and A. Fred, eds.), vol. 2, pp. 325–330, SciTePress, 2012.
- [16] E. Emerson, *Handbook of Theoretical Computer Science*, vol. B, ch. Temporal and Modal Logic, pp. 995–1072. Elsevier, MIT Press, 1990.
- [17] F. Wolter and M. Wooldridge, Temporal and dynamic logic, *Journal of Indian Council of Philosophical Research*, vol. XXVII(1), pp. 249–276, 2011.
- [18] J. van Benthem, *Handbook of Logic in Artificial Intelligence and Logic Programming*, ch. Temporal Logic, pp. 241–350. 4, Clarendon Press, 1993–95.
- [19] M. d'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.
- [20] D. Riehle and H. Zullighoven, Understanding and using patterns in software development, *Theory and Practice of Object Systems*, vol. 2(1), pp. 3–13, 1996.

Software Bug Ontology Supporting Bug Search on Peer-to-Peer Networks

Ha Manh TRAN^{a,1}, Son Thanh LE^a, Synh Viet Uyen HA^a and Tu Kha HUYNH^a

^aComputer Science, International University - HCMC National University, Vietnam

Abstract. This paper presents a semantics-based bug search system that allows users to solve bugs by searching similar bug reports on peer-to-peer networks. This system uses a bug schema to obtain and format bug reports from different bug tracking systems. The bug schema modified from the existing bug schemas supports several properties for search purpose. The system also extends the Gnutella protocol to provide search operations on peers. We have implemented and experimented the prototyping system on a distributed computing testbed. The paper also includes performance evaluation and lessons learned.

Keywords. Bug Ontology, Bug Tracking System, Peer-to-Peer Network, Bug Search

1. Introduction

With the support of general purpose search engines, it becomes common practice for system administrator and network operators to look for fault solutions in indexed public archives. However, there are still a large number of the knowledge bases that these engines insufficiently take advantage of the data for search and index. Cloud systems, for example, fostering the centralization of various services require to build the central knowledge bases for the user support of fault resolution, or virtual communities including peer-to-peer (P2P) networks, decentralized social networks and online forums connect groups of experts to establish the expertise knowledge bases supporting fault resolution, such as solving bugs on open source packages.

Software bugs from bug tracking systems (BTS), network tickets from ticket tracking systems, problem discussions from forums, vendor knowledge bases, and virtual communities contain semi-structured information that usually cause difficulties for the search engines to exploit. The structured part contains fields and meta-data used to manage and classify bug reports. The simple classifications focus on packages, version, platform, severity, etc. The unstructured part contains pieces of text used to describe and discuss bug reports. This part exposes valuable data to conduct solutions for bug reports. The text mining based search methods usually deal with text processing techniques [1] leading to high computation cost as the amount of data increases. The keyword based search methods ignore to exploit the semantics of the data. This study uses a bug schema to represent and format bug reports, which exploit several types of classifications and

¹Corresponding Author: Dr. Ha Manh Tran, School of Computer Science and Engineering, International University - Ho Chi Minh City National University, Vietnam; E-mail: tmha@hcmiu.edu.vn

relationships. The underlying idea of this method is a problem occurring in a component can be caused by another problem occurring in a related component. At the first sight, both problems appear to be distinct, but share few similar symptoms and error messages. However, diagnosing processes finally find out the correlation of these problems.

We studies the existing bug schemas to propose a bug schema that exploits several types of classifications and relationships for search purpose. The bug schema is then applied to obtain bug reports on a P2P bug search system. The contribution is thus four-fold: (i) proposing a unified bug schema for a semantics-based bug search method; (ii) crawling and formatting bug reports using the defined bug schema; (iii) extending the Gnutella protocol [2] to implement a P2P bug search system; (iv) experimenting the P2P bug search system on a distributed computing testbed.

The rest of the paper is structured as follows: the next section introduces some bug resolution systems and existing bug schemas. Section 3 describes a bug schema extended to support for search purpose and presents crawling and formatting bug reports from multiple BTSs. Section 4 depicts a P2P bug search system built on the extended Gnutella protocol, the bug schema and the bug database. The performance evaluation of the system is reported in Section 5 with explanation before the paper is concluded in Section 6.

2. Related Work

While no popular BTS known to date already allows for retrieving bug data in resource description framework (RDF) and utilizing semantic links, this has been explored in earlier research prototypes. A Dhruv system [3] supports bug resolution in open source software communities by interlinking code artifacts, bug reports, discussion posts, co-authorship and community members, and then recommending related resources. We put more emphasis on packages, dependencies, symptoms and classifications. Schügerl et al. have extended an existing software engineering ontology to representing bug reports, particularly describing their *quality* [4]. The assessment of the quality relies on analyzing the natural language descriptions of bug reports. Evoont [5] is a collection of software, version and bug ontologies for software engineering. It serves software design, software analysis and bug tracking purposes. Baetle (Bug And Enhancement Tracking LanguagE) [6] and Helios [7] are work in progress towards a unified ontology for software bugs. The former aims to enable complex queries on bug data, explore the dependency of bug reports and exchange bug data among BTSs. Whereas the later proposes an open source application lifecycle management platform that deals with the synchronization and traceability issues of bug reports and the interoperability of bug data in BTSs.

We choose not to adopt these complex designs because we focus on analyzing bug semantics to search for similar bug reports. Two bug reports, for example, specifying “Firefox cannot connect to the Internet after installing updates” and “Cannot connect to the Internet after upgrading ZoneAlarm” possibly possess the same root cause of fire-wall blocked connections, even their descriptions do not seem to be similar. We however consider adopting and extending a subset of these schemas for this study. The extended schema contains multiple attributes that reasonably examine similar bug reports by different categories. The two bug report above can be classified by the groups of “connection failed” and “software updated”. Among several processes of crawling, grouping, and querying bug reports, the grouping process requires field-value pairs, key-words and

group-words detailed by the extended schema. The existing bug ontologies do not seem to expose this information, rather they only offer a unified schema for storing various types of bug reports without fostering similarity search capabilities. The querying process uses SPARQL queries to determine the similarity between the query and the bug reports formatted by the schema. The extension of SPARQL, namely iSPARQL, provides join operators to match similar code for analyzing software projects [8].

This study is an improvement of the retrieval component of the DisCaRia system [9], a distributed case-based reasoning system used for fault resolution of network and communication systems. DisCaRia applies the conventional case-based reasoning approach to explore problem solving knowledge resources on the P2P networks, where peers work as agents to share fault resources and coordinate fault solving activities. The current retrieval component applies text processing techniques to process the large bug database, causing high computation cost on peers.

3. Bug Data

We consider to exploit bug solution knowledge from several bug tracking systems that possess various bug schemas, and to construct the relationships and classifications of bug reports. We need a unified bug schema for crawling and formatting bug reports.

3.1. Unified Bug Schema

A unified bug schema allows various types of bug reports to be stored in one database, and also allows peers to interoperate in collaborative activities explained in the next section. Evoont [5], Baetle [6] and Helios [7] have proposed three bug schemas *bom*, *baetle* and *helios_bt*, respectively. The *bom* schema shares several common entities with the *baetle* schema. They use workflow ontology (wf) [10] to control tasks and states for bug reports. The *bom* schema maintains the resolution entity separate from the bug entity, while the *baetle* schema combines them together. A bug report following the *bom* schema occurs in a component of a product that belongs to a computer system or a project. It contains attachments, comments and activities from related users identified by semantically interlinked online communities (sioc) [11]. A bug report following the *baetle* schema is associated with a software package or a project defined by description of a project (doap) [12]. It also contains posts, comments and activities from related users who possess online accounts described by friend of a friend (foaf) [13]; especially, it can be classified by concepts defined by simple knowledge organisation systems (skos) [14]. The *helios_bt* schema extends the *bom* and *baetle* schemas to including the bug tracker entity that collects bug reports for a project. Except for the *baetle* schema that contains concept based classifications used for searching similar bug reports, the other schemas are only used for storing and relating bug reports.

Resource description framework (RDF) [15] has been widely used for representing information about Web resources; particularly representing meta-data information for Web resources and information about objects identified on the Web [16]. We use RDF to represent and correlate bug reports. The unified bug schema shown in Figure 1 describes bug entity, bug properties and other related entities. Several properties and entities are inherited from existing ontologies like sioc, doap, dc [17] (Dublin Core) and XML Schema

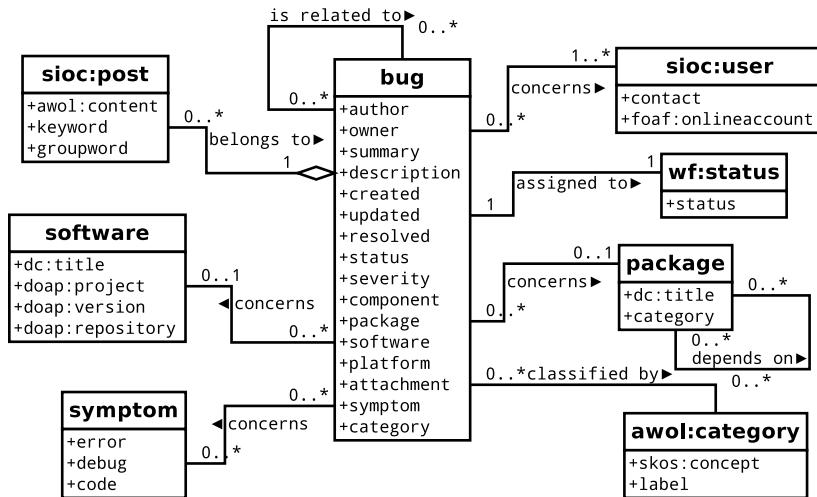


Figure 1. Unified bug schema represented as a UML diagram

Data Types [18]. The bug entity contains several properties that facilitate figuring out the relationship among bug reports, for example, bug reports are concerned with packages and components, bug reports are classified by symptoms and categories, or bug reports provide the dependency and relationship of other bug reports.

3.2. Bug Crawling

Some BTSSs support XML-RPC web service interface for users to access and modify bug reports. Bugzilla XML-RPC interface, for example, allows users to obtain a large number of bug reports in a single request. This interface contains three main modules: user module to manage user accounts, bug module to manage bug reports and product module to provide the information of products. However, the interface restricts users from obtaining the attachments and related discussion entries of bug reports. It thus cannot be suitable for our schema that requires this information for searching. We have implemented a bug crawler using XML-RPC web service interface for the purpose of updating bug reports because this interface performs efficiently on detecting any update on bug reports.

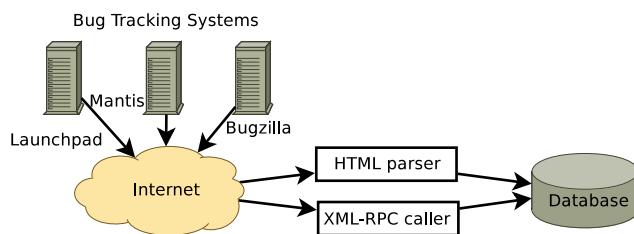


Figure 2. Architecture of the bug crawler

Several other BTSSs only allow users to access and modify bug reports through HTML-based web interface. Obtaining bug reports using this interface is rather slow and

complex because HTML pages usually lack consistency and the same structures can be presented in different ways. In order to parse bug reports, the parser of a bug crawler takes advantage of the bug template that provides the same structure for bug reports within a BTS. Different BTSs possess different bug templates and appearances, for example, Bugzilla, Launchpad, Mantis templates. The bug crawler as shown in Figure 2 contains multiple modules: downloading module to obtain the HTML pages of bug reports, parsing module to extract the content of bug reports, storing module to save the extracted content to database using the unified bug schema.

3.3. Bug Formatting

While bug properties can be extracted directly from parsing bug reports, package dependency, bug dependency, bug relationship, and bug classification require additional processing effort. Bug reports belong to packages and packages depend on other packages, bug dependency thus considers sets of bug reports from the same packages and the dependent packages. Some bug schemas directly provide dependent bug reports. Bug relationship can be obtained by matching patterns on discussions, e.g., regular expression. Bug identities appearing in a bug report's discussions are more or less related to the bug report. The relationship set contains these bug identities. Bug error messages can also be obtained by matching patterns on discussions, e.g., service unavailable, not found, network is unreachable, etc.

Classifying bug reports relies not only on above information but also on distinct keywords and groups of keywords. We have used the term frequency - inverse document frequency ($tf \times idf$) method to assess the significance of keywords to bug reports in the bug database. The $tf \times idf$ value of a keyword depends on the occurrence frequency of the keyword in a bug report over the total number of keywords of the bug report (term frequency) and the occurrence frequency of the keyword in other bug reports over the total number of bug reports (inverse document frequency). A keyword set contains keywords with high $tf \times idf$ values and a keyword group contain a keyword set of a certain discussion. Formatting bug reports is a rather engineering process that contains keyword filter, assessment, selection and category.

4. P2P Bug Search System

P2P networks have been largely applied for multimedia file sharing applications [19,20] due to several good features of self-management, high scalability and flexibility in distributed environment. These networks can also be used for retrieval and searching applications [21,22]. Except for the above good features, there are apparently additional advantages when exploiting P2P networks for these applications. First, they can exploit domain-specific data on virtual communities; especially, search queries can be addressed by groups of experts in order to provide good results. Second, they can avoid processing a large amount of data that usually consumes a lot of computing resources. The data is scattered on a collection of workstations without administration, rather than on a single server or a group of servers with administration. Moreover, they can exploit the computing and storage capability of each workstation on P2P networks.

We have developed a P2P bug search system based on the Gnutella networks [2]. There are two types of peers in the Gnutella networks. A peer possesses normal process-

ing capability, while a super peer possesses high processing capability and high availability without being behind a firewall. Each peer only connects to one of the super peers, and the super peers connect to each other and to peers. The super peers also act as proxies to the Gnutella network for peers. Queries are forwarded among the super peers using various routing mechanisms. When receiving the queries, the super peers only forward the queries to peers that match the queries' search keys. Peers become the super peers if they can prove their capability. The super peers resolve the heterogeneity problem and increase the scalability of the Gnutella network by reducing the number of incapable peers incorporating in query routing mechanisms.

The Gnutella protocol supports five types of messages: *ping* and *pong* used to probe the network, *query* and *queryhit* used to exchange data, and *push* used to deal with peers behind the firewall. A Gnutella message consists of header and content. The attributes of the header are shown as follows:

```
Original Gnutella message header
+-----+-----+-----+-----+
| message id | descriptor | ttl | hops | payload length |
+-----+-----+-----+-----+
```

The *message id* field is used to detect whether a message has already arrived at a certain peer before. The *payload descriptor* field indicates the type of a message such as, ping (0×00), pong (0×01), query (0×80), queryhit (0×81) and push (0×40). The *ttl* field is the number of times that a message can be forwarded in the network while the *hops* field is the number of times that a message has been forwarded. The *payload length* field is the in-byte size of the content that immediately follows the header. The detailed structure of the content depending on message types is defined in the protocol specification [2].

```
Original Gnutella query message
+-----+-----+
| minspeed(kb/s) | query string |
+-----+-----+
```

```
Original Gnutella queryhit message
+-----+-----+-----+-----+-----+
| numhits | port | ipaddress | speed(kb/s) | resultset | identity |
+-----+-----+-----+-----+-----+
```

The P2P bug search system uses the super peers to perform the search operation. The Gnutella protocol is therefore extended to enable this operation on the super peers. The modification of two query and queryhit messages is rather straightforward. The query message contains an SPARQL query [23], while the queryhit message contains a set of results that is usually large in size and complex in structure. Since this system receives the resulting sets directly in the queryhit messages, the download process is ignored. The information of port and IP address is redundant, while the information of minimum speed and speed discloses the data transmission rate of the super peers.

Figure 3 plots the architecture of the super peer of the system. Peer controller communicates other peers in the P2P network to receive queries and send queryhits. Query handler performs SPARQL queries on the database provided the bug schema, then returns the resulting set to the peer controller. This handler also contains a module to process SPARQL queries; i.e., this module can traverse the RDF graph constructed by the

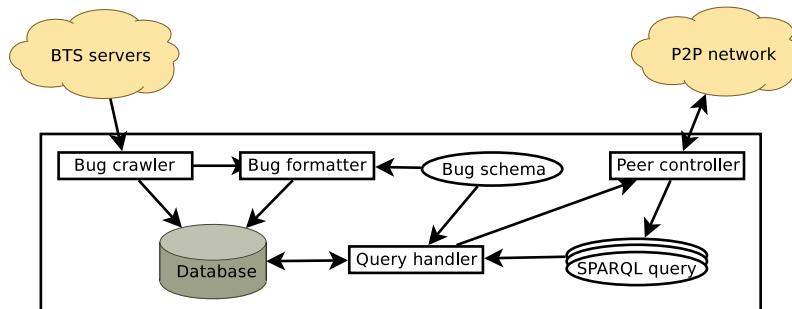


Figure 3. Architecture of the super peer

database to provide results for the queries. On the other hand, upon obtaining bug reports, bug crawler can either send to the bug formatter or update directly to the database. Bug formatter uses the bug schema to format and store bug reports in the database. Moreover, some super peers support a web-client module associated with a web server that allows users to connect to the system easily through web browsers.

Table 1. Number of bug reports obtained from BTS sites

BTS Site	BTS System	Number of Bugs
bugs.debian.org	Debian BTS	300000
bugs.eclipse.org	Bugzilla	150000
bugs.gentoo.org	Bugzilla	200000
bugzilla.mozilla.org	Bugzilla	300000
bugzilla.redhat.com	Bugzilla	200000
issues.asterisk.org	Mantis	20000
bugs.scribus.net	Mantis	10000
bugs.launchpad.net/ubuntu	Launchpad	200000

5. Evaluation

We have crawled bug reports from several BTSs for the P2P bug search system. Table 1 reports the number of bug reports obtained from different BTS sites. The search method discussed in this study uses the information of bug relationship, dependency and several types of classifications to retrieve similar bug reports on the P2P network. With the support of the bug schema, the method uses SPARQL queries to query the well formatted bug database on peers. We consider two metrics: identical bug reports and similar bugs reports. Two bug reports are similar if they fall in the same type of classifications, i.e., matching keywords, groups of keywords or symptoms. The method first filters bug reports in the same packages, the dependent packages, the relationship sets, and then examines their classifications.

We have used EMANICSLab (European Network of Excellence for the Management of Internet Technologies and Complex Services) to perform experiments. EMANICSLab is a flexible and highly re-usable distributed computing and storage testbed to

support joint research activities of EMANICS partners. Figure 4 plots the topology of the P2P bug search system, where super peers locate on EMANICSLab's nodes. The system supports two types of clients for connecting to the super peer network. Users can use either peers to connect directly to one of the super peers or web browsers to connect indirectly to the super peer through its associated web server.

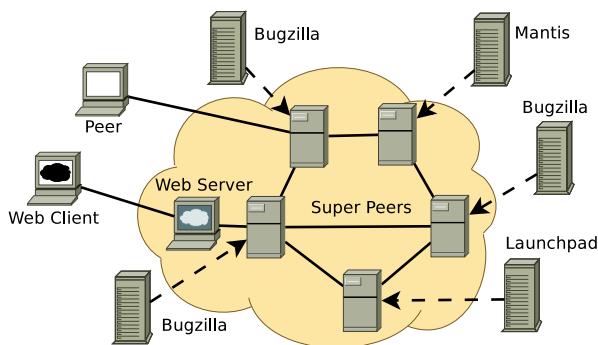


Figure 4. Topology of the P2P bug search system

One of the motivations to build this system is to exploit resources usage at super peers, i.e., an ordinary workstation with sufficient resource capability can become a super peer. The first experiment measures the memory usage of a super peer for different bug datasets. Bug reports usually possess different sizes ranging from few kilobytes to few megabytes depending on the number of discussion posts. A super peer needs 200 megabytes RAM memory on average to load 100000 bug reports as shown in Figure 5 (left), thus normal hardware configurations, such as Intel Pentium Dual-Core Processor 1.3 GHz with 2GB RAM, possibly accommodate the reasonable number of bug reports.

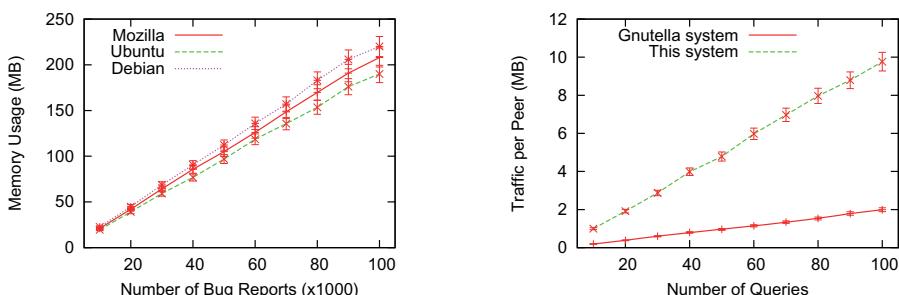


Figure 5. Memory usage for different bug datasets (left). Traffic comparison between the Gnutella systems and the P2P bug search system (right)

Unlike the Gnutella P2P systems, this system uses the queryhit messages to carry the resulting sets, ignoring the download process. The network traffic generated therefore increases on the P2P network considerably as the number of queries increases. The

second experiment compares the traffic generated on the Gnutella systems with this system using different query sets, as shown in Figure 5 (right). We have set 50 most relevant bug reports for each resulting set, while the Gnutella queryhit messages contain fewer results than 50 on average. Moreover, the content of each bug report is much more than the content of the Gnutella results that only specifies file names for the Gnutella file sharing systems. The traffic generated on this system is therefore larger than on the Gnutella systems. This is one of the reasons that we consider using super peers for this system.

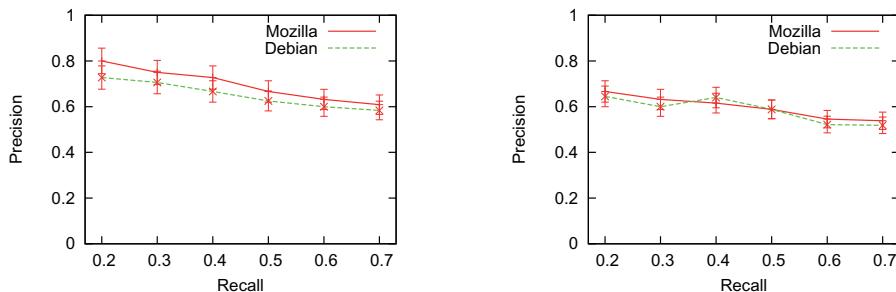


Figure 6. Precision rate over recall rate for the identical bug reports (left). Precision rate over recall rate for the similar bug reports (right)

We have created a query set to evaluate the identical bug reports. Queries in the query set are selected from the bug databases that contain overlapping sets of bug reports intentionally, e.g., super peers possess Mozilla bug reports ranging from id=50000 to id=150000 and from id=100000 to id=200000. The third experiment evaluates the precision of the search method by retrieving the number of the identical bug reports defined. Figure 6 (left) presents precision rate over recall rate using the query set of 100 queries. The search method performs better with the Mozilla database than the Debian database. Bug reports in the Debian database belong to too many packages, causing a small number of bug reports found in the same packages, the dependent packages and the relationship sets.

Evaluating similar bug reports is rather difficult because bug databases are wide in scope and remain largely unexplored. We have customized the query set in the third experiment by removing some keywords and symptoms from queries. The resulting sets of the identical bug reports remain unchanged. The fourth experiment evaluates the precision of the search method by retrieving a number of the similar bug reports that are actually the identical bug reports. The search method performs similarly with both Mozilla and Debian databases. The number of similar bug reports retrieved increases as queries possess fewer keywords and symptoms. However, precision rate is reduced because several similar bug reports are unrelated to the queries, as shown in Figure 6 (right).

6. Conclusions

We have proposed and implemented the semantics-based bug search system on the P2P network. This system uses the unified bug schema to represent and format bug reports obtained from different BTSSs. The bug schema extended from the existing bug schemas

supports several types of classifications and relationships for the purpose of searching similar bug reports. The system also employs a P2P architecture where super peers possess sufficient processing and storage capability to accommodate large bug databases and perform SPARQL queries on the databases. We have extended the Gnutella P2P protocol to enable search capability on super peers. The experimental evaluation of the system on EMANICSLab reveals two remarkable issues: the scalability and efficiency of the system. Even super peers possess reasonable bandwidth for data transfer, the traffic generated on the P2P network increases significantly as the number of queries increases. The evaluation of similar bug reports is yet incomplete because bug databases are wide in scope and remain largely unexplored. Future work will address these issues.

Acknowledgment This research work is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2011.01.

References

- [1] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science and Technology*, 41(6), 391–407, 1990.
- [2] Gnutella Protocol Specification version 0.4. <http://rfc-gnutella.sourceforge.net/developer/stable/>. Last access in Jan. 2013.
- [3] A. Ankolekar, K. Sycara, J. Herbsleb, R. Kraut, C. Welty. Supporting Online Problem-Solving Communities with the Semantic Web. In *Proc. 15th International Conference on World Wide Web (WWW06)*, pp. 575–584, ACM, 2006.
- [4] P. Schügerl, J. Rilling, P. Charland. Enriching SE Ontologies with Bug Report Quality. In *Proc. 4th International Workshop on Semantic Web Enabled Software Engineering (SWESE 08)*, 2008.
- [5] EvoOnt - Software Evolution Ontology. <https://filesifi.uzh.ch/ddis/oldweb/ddis/research/evoont/>. Last access in Jan. 2013.
- [6] Bug and Enhancement Tracking Language. <http://code.google.com/p/baetle/>. Last access in Jan. 2013.
- [7] Bug Ontology. http://heliosplatform.sourceforge.net/ontologies/helios_bt.html. Last access in Jan. 2013.
- [8] C. Kiefer, A. Bernstein, and J. Tappolet. Analyzing Software with iSPARQL. In *Proc. 3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE 07)*, 2007.
- [9] H. M. Tran and J. Schönwälder. Distributed Case-Based Reasoning for Fault Management. In *Proc. 1st Intl. Conf. on Autonomous Infrastructure, Management and Security*, pp. 200–203, Springer, 2007.
- [10] Workflow Ontology. <http://code.google.com/p/baetle/wiki/WorkFlowOntology>. Last access in Jan. 2013.
- [11] Semantically-Interlinked Online Communities (SIOC). <http://sio-project.org/>. Last access in Jan. 2013.
- [12] Description of a Project (DOAP). <https://github.com/edumbill/doap/wiki>. Last access in Jan. 2013.
- [13] Friend Of A Friend (FOAF). <http://www.xmlns.com/foaf/spec/>. Last access in Jan. 2013.
- [14] Simple Knowledge Organization System. <http://www.w3.org/2004/02/skos/>. Last access in Jan. 2013.
- [15] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Last access in Jan. 2013.
- [16] K. K. Breitman and M. A. Casanova and W. Truszkowski. *Semantic Web: Concepts, Technologies and Applications*, Springer, 2007.
- [17] Dublin Core Metadata Initiative. <http://dublincore.org/>. Last access in Jan. 2013.
- [18] P. V. Biron, K. Permanente, and A. Malhotra. XML Schema Part 2: Datatypes Second Edition. <http://www.w3.org/TR/xmlschema-2/>. Last access in Jan. 2013.
- [19] Napster. <http://www.napster.com/>. Last access in Jan. 2013.
- [20] B. Cohen. Incentives Build Robustness in BitTorrent. In *Proc. 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [21] Faroo. <http://www.faroo.com/>. Last access in Jan. 2013.
- [22] Yacy. <http://www.yacy.de/>. Last access in Jan. 2013.
- [23] D. D. Chamberlin and R. F. Boyce. SEQUEL: A Structured English Query Language. In *Proc. ACM SIGFIDET Workshop on Data Description, Access and Control*, pp. 249–264, ACM, 1974.

Agent Theories, Models and Communication

On the Horn Fragments of Serial Regular Grammar Logics with Converse

Linh Anh NGUYEN^{a,b,1}, and Andrzej SZALAS^{a,c}

^a Institute of Informatics, University of Warsaw, Poland

^b VNU University of Engineering and Technology, Hanoi, Vietnam

^c Dept. of Computer and Information Science, Linköping University, Sweden

Abstract. We study Horn fragments of serial multimodal logics which are characterized by regular grammars with converse. Such logics are useful for reasoning about epistemic states of multiagent systems as well as similarity-based approximate reasoning. We provide the first algorithm with PTIME data complexity for checking satisfiability of a Horn knowledge base in a serial regular grammar logic with converse.

Keywords. grammar logics, epistemic logics, rule languages

1. Introduction

Horn fragments of applied logics have attracted lot of attention. Rules in the form of implication $B_1 \wedge \dots \wedge B_k \rightarrow A$ are widely used in practice of logic programming, deductive databases and knowledge representation and reasoning. Moreover, Horn fragments typically enjoy PTIME data complexity, in contrast to (at least) NP-hardness of full logics.

In [1] Nguyen studied the Horn fragment of serial regular grammar logics without converse. He gave an algorithm that, given a positive multimodal logic program P and a set of finite automata specifying a serial regular grammar logic L , constructs a finite least L -model of P . A least L -model M of P has the property that for every positive formula φ , $P \models \varphi$ iff $M \models \varphi$. The algorithm presents a bottom-up method for checking satisfiability of a Horn knowledge base in a serial regular grammar logic.

Extending the method and results of [1] for serial regular grammar logics with converse is a challenging task. As mentioned in [1], when L is the symmetric modal logic KDB or B , a least L -model of the simple positive modal logic program $\{p\}$ exists but it must be infinite.²

In this paper, we study Horn fragments of serial regular grammar logics with converse. Similarly to [1,2], we assume that the corresponding grammars of such

¹ Corresponding Author: Linh Anh Nguyen, Institute of Informatics, University of Warsaw, Banacha 2, 02-097, Warsaw, Poland; E-mail: nguyen@mimuw.edu.pl.

² It also seems that there are positive modal logic programs in serial regular grammar logics with converse that do not have least models (w.r.t. positive consequences).

logics are specified by finite automata. Following the approach of description logics [3,4], we define a Horn knowledge base to be a pair $\langle \Gamma, \mathcal{A} \rangle$, where Γ is a finite set of Horn formulas (consisting of positive program clauses and negative clauses) and \mathcal{A} is a finite set of assertions about states, called an ABox. As the main contribution we provide the first algorithm with PTIME data complexity for checking satisfiability of a Horn knowledge base in a serial regular grammar logic with converse, where data complexity is measured w.r.t. the size of the ABox \mathcal{A} and it is assumed that the logic and Γ are fixed. Note that the combined complexity of the considered satisfiability problem, measured w.r.t. the sizes of \mathcal{A} , Γ and the automata specifying the logic, is conjectured by Nguyen [1] to be EXPTIME-complete even for serial regular grammar logic without converse.

Our method extends the method of Nguyen [1] by using a special technique for dealing with converse and by directly checking satisfiability of a given Horn knowledge base without constructing a least model for the positive part of the Horn knowledge base. Unlike in [1], it is not bottom-up and can be made top-down by applying an appropriate search strategy.

The rest of this paper is structured as follows. In Section 2 we give definitions for serial regular grammar logics with converse and the Horn fragments of these logics. We also recall automaton-modal operators [1,4,5]. Section 3 presents our algorithm. Section 4 concludes this work. Due to the lack of space, proofs of our results are only presented in the long version [6] of the current paper.

2. Preliminaries

2.1. Grammar Logics

Grammar logics [1,2,4,5,7,8,9,10] are normal multimodal logics characterized by “inclusion” axioms of the form $[\sigma_1] \dots [\sigma_h]\varphi \rightarrow [\varrho_1] \dots [\varrho_k]\varphi$. Such axioms correspond to grammar rules $\sigma_1 \dots \sigma_h \rightarrow \varrho_1 \dots \varrho_k$ when modal indices are treated as grammar symbols. A grammar logic L is called a *context-free grammar logic* if the corresponding grammar, denoted by $\mathcal{G}(L)$, is context-free. It is called a *regular grammar logic* if it is context-free and, for every modal index σ , the set of words $\mathcal{G}(L)$ -derivable from σ is a regular language. A *context-free grammar logic with converse* is a context-free grammar logic L such that if $\sigma \rightarrow \varrho_1 \dots \varrho_k$ is a grammar rule of $\mathcal{G}(L)$ then $\bar{\sigma} \rightarrow \bar{\varrho}_k \dots \bar{\varrho}_1$ is also a grammar rule of $\mathcal{G}(L)$, where $\bar{\sigma}$, $\bar{\varrho}_k$, \dots , $\bar{\varrho}_1$ are modal indices, called *converses* of σ , ϱ_k , \dots , ϱ_1 , respectively. A *serial grammar logic* is a grammar logic extended by seriality axioms $\langle \sigma \rangle \top$, for all modal indices σ . Thus, a *serial regular grammar logic with converse* is a context-free grammar logic with converse that is regular and serial.

The class of serial regular grammar logics with converse contains useful epistemic logics. The seriality axioms (written in the form $[\sigma]\varphi \rightarrow \neg[\sigma]\neg\varphi$) state that beliefs of an agent (or a group of agents) σ are consistent, while inclusion axioms can be used, for example, to express positive introspection of knowledge and belief ($[\sigma]\varphi \rightarrow [\sigma][\sigma]\varphi$) or to represent knowledge sharing between agents or groups of agents (e.g., $[\sigma]\varphi \rightarrow [\varrho]\varphi$ or $[\sigma]\varphi \rightarrow [\varrho][\sigma]\varphi$). With converse, negative introspection of knowledge and belief ($\neg[\sigma]\varphi \rightarrow [\sigma]\neg[\sigma]\varphi$) can also be expressed by using

the grammar rule $\sigma \rightarrow \bar{\sigma}\sigma$. For applications and a methodology of constructing such logics see, e.g., [9,10].

Apart from reasoning about epistemic states of agents and groups of agents, serial regular grammar logics with converse can also be used for similarity-based approximate modeling and reasoning. In this case, instead of states and accessibility relations, one deals with objects and similarity relations [4,11,12].

2.2. Regular Semi-Thue Systems

Let Σ_s , Σ_p , Σ_{m+} be non-empty, pairwise disjoint finite sets of *state names*, *propositions*, and *modal indices*, respectively. For $\sigma \in \Sigma_{m+}$, by $\bar{\sigma}$ we denote a fresh symbol, called the *converse* of σ . We use notation $\Sigma_{m-} = \{\bar{\sigma} \mid \sigma \in \Sigma_{m+}\}$ and assume that Σ_{m-} is disjoint with Σ_{m+} , Σ_s and Σ_p . For $\varrho = \bar{\sigma} \in \Sigma_{m-}$, by writing $\bar{\varrho}$ we mean σ . We call $\Sigma_m = \Sigma_{m+} \cup \Sigma_{m-}$ an *alphabet with converse* (for modal indices).

A *context-free semi-Thue system* S over Σ_m is a finite set of context-free production rules over alphabet Σ_m . We say that S is *symmetric* if, for every rule $\sigma \rightarrow \varrho_1 \dots \varrho_k$ of S , the rule $\bar{\sigma} \rightarrow \bar{\varrho}_k \dots \bar{\varrho}_1$ is also in S .

A context-free semi-Thue system is like a context-free grammar, but it has no designated start symbol and there is no distinction between terminal and non-terminal symbols. We assume that for $\sigma \in \Sigma_m$, the word σ is derivable from σ using such a grammar.

A context-free semi-Thue system S over Σ_m is called a *regular semi-Thue system* S over Σ_m if, for every $\sigma \in \Sigma_m$, the set of words derivable from σ using the system is a regular language over Σ_m .

Similarly to [2], we assume that any considered regular semi-Thue system S is always given together with a mapping \mathbf{A} that associates each $\sigma \in \Sigma_m$ with a finite automaton \mathbf{A}_σ recognizing words derivable in S from σ . We call \mathbf{A} the *mapping specifying the finite automata of S*. Note that it is undecidable to check whether a context-free semi-Thue system is regular [13].

Recall that a *finite automaton* \mathbf{A} over alphabet Σ_m is a tuple $\langle \Sigma_m, Q, q_0, \delta, F \rangle$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta \subseteq Q \times \Sigma_m \times Q$ is the transition relation, and $F \subseteq Q$ is the set of accepting states. A *run* of \mathbf{A} on a word $\sigma_1 \dots \sigma_k$ is a finite sequence of states q_0, q_1, \dots, q_k such that $\delta(q_{i-1}, \sigma_i, q_i)$ holds for every $1 \leq i \leq k$. It is an *accepting run* if $q_k \in F$. We say that \mathbf{A} *accepts* a word w if there exists an accepting run of \mathbf{A} on w .

2.3. Serial Regular Grammar Logics with Converse

Let $\Sigma = \Sigma_s \cup \Sigma_p \cup \Sigma_m$ and $\Sigma_+ = \Sigma_s \cup \Sigma_p \cup \Sigma_{m+}$. We refer to elements of Σ_{m-} also as *modal indices*. We use letters like a , b to denote state names, letters like p , q to denote propositions, and letters like σ , ϱ to denote modal indices.

In the *base language*, *formulas* over Σ (respectively Σ_+) are defined by the following BNF grammar, where $p \in \Sigma_p$ and $\sigma \in \Sigma_m$ (respectively $\sigma \in \Sigma_{m+}$):

$$\varphi, \psi ::= \top \mid \perp \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid [\sigma]\varphi \mid \langle \sigma \rangle \varphi$$

An *interpretation* over Σ is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set of states, called the *domain* of \mathcal{I} , and $\cdot^{\mathcal{I}}$ is a function, called the *interpreta-*

Table 1. Interpretation of complex formulas.

$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$	$\perp^{\mathcal{I}} = \emptyset$	$(\neg\varphi)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \varphi^{\mathcal{I}}$
$(\varphi \wedge \psi)^{\mathcal{I}} = \varphi^{\mathcal{I}} \cap \psi^{\mathcal{I}}$	$(\varphi \vee \psi)^{\mathcal{I}} = \varphi^{\mathcal{I}} \cup \psi^{\mathcal{I}}$	$(\varphi \rightarrow \psi)^{\mathcal{I}} = (\neg\varphi \vee \psi)^{\mathcal{I}}$
$([\sigma]\varphi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \langle x, y \rangle \in \sigma^{\mathcal{I}} \rightarrow y \in \varphi^{\mathcal{I}}\}$		
$(\langle\sigma\rangle\varphi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \langle x, y \rangle \in \sigma^{\mathcal{I}} \wedge y \in \varphi^{\mathcal{I}}\}.$		

tion function, that maps each state name $a \in \Sigma_s$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each proposition $q \in \Sigma_p$ to a set $q^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each modal index $\sigma \in \Sigma_{m+}$ to a binary relation $\sigma^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For $\sigma \in \Sigma_{m+}$, we define $\bar{\sigma}^{\mathcal{I}} = (\sigma^{\mathcal{I}})^{-1}$. The interpretation function is extended to complex formulas over Σ as in Table 1.

For a set Φ of formulas, we define $\Phi^{\mathcal{I}} = \bigcap\{\varphi^{\mathcal{I}} \mid \varphi \in \Phi\}$.

An *ABox* over Σ is a finite set of *assertions* of the form $p(a)$, $\neg p(a)$ or $\sigma(a, b)$, where $p \in \Sigma_p$, $a, b \in \Sigma_s$ and $\sigma \in \Sigma_{m+}$. A *knowledge base* over Σ (respectively Σ_+) is a pair $\langle \Gamma, \mathcal{A} \rangle$, where Γ is a finite set of formulas over Σ (respectively Σ_+) and \mathcal{A} is an *ABox* over Σ . The formulas of Γ are called the *global assumptions* of the knowledge base.

An interpretation \mathcal{I} *validates* a global assumption φ if $\varphi^{\mathcal{I}} = \Delta^{\mathcal{I}}$. It *validates* an assertion $p(a)$ (respectively $\neg p(a)$, $\sigma(a, b)$) if $a^{\mathcal{I}} \in p^{\mathcal{I}}$ (respectively $a^{\mathcal{I}} \notin p^{\mathcal{I}}$, $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in \sigma^{\mathcal{I}}$). An interpretation is a *model* of a knowledge base $\langle \Gamma, \mathcal{A} \rangle$ if it validates all global assumptions of Γ and all assertions of \mathcal{A} .

Given two binary relations r and s , their relational composition is defined by $r \circ s = \{\langle x, y \rangle \mid \exists z \langle x, z \rangle \in r \wedge \langle z, y \rangle \in s\}$.

Let S be a symmetric regular semi-Thue system over Σ_m . An interpretation \mathcal{I} over Σ is called an $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) *interpretation* if:³

- for every rule $\sigma \rightarrow \varrho_1 \dots \varrho_k$ of S , $\varrho_1^{\mathcal{I}} \circ \dots \circ \varrho_k^{\mathcal{I}} \subseteq \sigma^{\mathcal{I}}$; and
- for every $\sigma \in \Sigma_m$ (respectively $\sigma \in \Sigma_{m+}$) and every $x \in \Delta^{\mathcal{I}}$, we have that $\{y \mid \langle x, y \rangle \in \sigma^{\mathcal{I}}\} \neq \emptyset$.

An $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) interpretation is called an $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) *model* of a knowledge base KB if it is a model of KB . A knowledge base is said to be *satisfiable* in the $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) logic if it has an $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) model.

We say that a state name a *satisfies* the property φ w.r.t. a knowledge base KB in the $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) logic, denoted by $KB \models_{SL_{\Sigma, S}} \varphi(a)$ (respectively $KB \models_{SL_{\Sigma_+, S}} \varphi(a)$), if, for every $SL_{\Sigma, S}$ (respectively $SL_{\Sigma_+, S}$) model \mathcal{I} of KB , we have that $a^{\mathcal{I}} \in \varphi^{\mathcal{I}}$.

2.4. Horn Fragments of the Considered Logics

A formula is called a *positive formula* if it does not contain \rightarrow , \neg , \perp . A formula is called a *negative formula* if it is $\neg\varphi$ for some positive formula φ . *Horn formulas* are inductively defined by:

³ SL in $SL_{\Sigma, S}$ and $SL_{\Sigma_+, S}$ stands for “serial (multimodal) logic”.

- \top , a proposition, a negative formula,
- $[\sigma]\varphi, \langle\sigma\rangle\varphi, \varphi \wedge \psi$, where φ and ψ are Horn formulas,
- $\varphi \vee \psi$ or $\psi \vee \varphi$, where φ is a negative formula and ψ is a Horn formula.

A knowledge base $\langle\Gamma, \mathcal{A}\rangle$ is called a *Horn knowledge base* if Γ consists of *Horn formulas*. The proof of the following proposition is trivial.

Proposition 1 *Given a Horn knowledge base $\langle\Gamma, \mathcal{A}\rangle$ over Σ (respectively Σ_+), a positive formula φ over Σ (respectively Σ_+) and $a \in \Sigma_s$, we have that $\langle\Gamma, \mathcal{A}\rangle \models_{SL_{\Sigma,S}} \varphi(a)$ (respectively $\langle\Gamma, \mathcal{A}\rangle \models_{SL_{\Sigma+,S}} \varphi(a)$) iff the Horn knowledge base $\langle\Gamma \cup \{\neg\varphi \vee p\}, \mathcal{A} \cup \{\neg p(a)\}\rangle$ is unsatisfiable in $SL_{\Sigma',S}$ (respectively $SL_{\Sigma'_+,S}$), where p is a fresh proposition and Σ' extends Σ with p .*

A *modal context* is a (possibly empty) sequence $[\sigma_1] \dots [\sigma_h]$. A *Horn clause* is a formula of the form

$$\Box(A_1 \wedge \dots \wedge A_k \rightarrow B) \text{ or } \Box(A_1 \wedge \dots \wedge A_k \rightarrow \perp),$$

where \Box is a modal context, $k \geq 0$ and A_1, \dots, A_k, B are formulas of the form p , $[\sigma]p$ or $\langle\sigma\rangle p$. Note that every Horn clause can be represented and hence treated as a Horn formula.

Proposition 2 *For any Horn knowledge base $\langle\Gamma, \mathcal{A}\rangle$ over Σ (respectively Σ_+), there exists a Horn knowledge base $\langle\Gamma', \mathcal{A}\rangle$ over some Σ' (respectively Σ'_+) such that Γ' consists of Horn clauses and $\langle\Gamma, \mathcal{A}\rangle$ is satisfiable in $SL_{\Sigma,S}$ (respectively $SL_{\Sigma+,S}$) iff $\langle\Gamma', \mathcal{A}\rangle$ is satisfiable in $SL_{\Sigma',S}$ (respectively $SL_{\Sigma'_+,S}$).*

This proposition can be proved by using the replacement technique as in [14].

The *length* of a formula is the number of symbols in its representation. By the *size* of a set of formulas we understand the sum of the lengths of its formulas. The *size* of an ABox is defined to be its cardinality.

By *data complexity class* of the Horn fragment of $SL_{\Sigma,S}$ (respectively $SL_{\Sigma+,S}$) we mean the complexity class of the problem of checking satisfiability of a Horn knowledge base $\langle\Gamma, \mathcal{A}\rangle$ in $SL_{\Sigma,S}$ (respectively $SL_{\Sigma+,S}$), measured in the size of \mathcal{A} , where it is assumed that Γ is a fixed set consisting of Horn clauses.

2.5. Automaton-Modal Operators

Given an interpretation \mathcal{I} and a finite automaton A over alphabet Σ_m , we define:

$$\mathsf{A}^{\mathcal{I}} = \{ \langle x, y \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \text{there exist a word } \sigma_1 \dots \sigma_k \text{ accepted by } \mathsf{A} \\ \text{and elements } x_0 = x, x_1, \dots, x_k = y \text{ of } \Delta^{\mathcal{I}} \text{ such that} \\ \langle x_{i-1}, x_i \rangle \in \sigma_i^{\mathcal{I}} \text{ for all } 1 \leq i \leq k \}$$

We will use auxiliary modal operators $[\mathsf{A}]$ and $\langle\mathsf{A}\rangle$, where A is a finite automaton over alphabet Σ_m . We call $[\mathsf{A}]$ (respectively $\langle\mathsf{A}\rangle$) a *universal* (respectively *existential*) *automaton-modal operator*.

In the *extended language*, if φ is a formula then $[\mathsf{A}]\varphi$ and $\langle\mathsf{A}\rangle\varphi$ are also formulas. The semantics of $[\mathsf{A}]\varphi$ and $\langle\mathsf{A}\rangle\varphi$ are defined as follows:

$$([A]\varphi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y(\langle x, y \rangle \in A^{\mathcal{I}} \text{ implies } y \in \varphi^{\mathcal{I}})\}$$

$$(\langle A \rangle \varphi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y(\langle x, y \rangle \in A^{\mathcal{I}} \text{ and } y \in \varphi^{\mathcal{I}})\}.$$

For a finite automaton A over Σ_m , assume that $A = \langle \Sigma_m, Q_A, q_A, \delta_A, F_A \rangle$.

If q is a state of a finite automaton A then by A_q we denote the finite automaton obtained from A by replacing its initial state by q .

3. Checking Satisfiability of Horn Knowledge Bases

Let S be a regular semi-Thue system specified by a mapping A and let L be either $SL_{\Sigma, S}$ or $SL_{\Sigma_+, S}$. In this section we present an algorithm that, given a Horn knowledge base $\langle \Gamma, A \rangle$ in L with Γ consisting of Horn clauses, checks whether $\langle \Gamma, A \rangle$ is satisfiable in L .

Let Σ_M stands for Σ_m when $L = SL_{\Sigma, S}$, and for Σ_{m+} when $L = SL_{\Sigma_+, S}$.

By $\Sigma_{\diamond}(\Gamma)$ we denote the set of elements $\sigma \in \Sigma_M$ such that a formula of the form $\langle \sigma \rangle p$ occurs at the left hand side of \rightarrow in some clause of Γ .

Let X be a set of formulas. The *saturation* of X (w.r.t. A and Γ), denoted by $\text{Sat}(X)$, is defined to be the least extension of X such that:

- if $[\sigma]\varphi \in \text{Sat}(X)$ then $[A_{\sigma}]\varphi \in \text{Sat}(X)$,
- if $[A]\varphi \in \text{Sat}(X)$ and $q_A \in F_A$ then $\varphi \in \text{Sat}(X)$,
- $\top \in \text{Sat}(X)$ and $\langle \sigma \rangle \top \in \text{Sat}(X)$ for every $\sigma \in \Sigma_M$,
- if $p \in \text{Sat}(X)$ and $\sigma \in \Sigma_{\diamond}(\Gamma)$ then $[A_{\bar{\sigma}}]\langle A_{\sigma} \rangle p \in \text{Sat}(X)$.

The *transfer* of X through $\langle \sigma \rangle$ is defined as:

$$\text{Trans}(X, \sigma) = \{[A_q]\varphi \mid [A]\varphi \in X \text{ and } \langle q_A, \sigma, q \rangle \in \delta_A\}.$$

Algorithm 1 checks the satisfiability of a given Horn knowledge base $\langle \Gamma, A \rangle$ in L with Γ consisting of Horn clauses. It uses the following data structures:

- Δ_0 : the set of all state names occurring in the ABox A .
In the case when A is empty, we set $\Delta_0 = \{\tau\}$.
- Δ : a set of states containing Δ_0 .
- *Label* : for every $x \in \Delta$, $\text{Label}(x)$ is a set of formulas called the *label* of x .
- *Next* : $\Delta \times \{\langle \sigma \rangle \top, \langle \sigma \rangle p \mid \sigma \in \Sigma_M, p \in \Sigma_p\} \rightarrow \Delta$ is a partial function with the meaning that $\text{Next}(x, \langle \sigma \rangle \varphi) = y$ holds when $\langle \sigma \rangle \varphi \in \text{Label}(x)$, $\varphi \in \text{Label}(y)$, and $\langle \sigma \rangle \varphi$ is “realized” at x by going to y .

Define $\text{Edges} = \{\langle x, \sigma, y \rangle \mid \sigma(x, y) \in A \text{ or } \text{Next}(x, \langle \sigma \rangle \varphi) = y \text{ for some } \varphi\}$.

We say that $x \in \Delta$ is *reachable* from Δ_0 (w.r.t. L) if there exist $x_0, \dots, x_k \in \Delta$ and $\sigma_1, \dots, \sigma_k \in \Sigma_M$ such that $k \geq 0$, $x_0 \in \Delta_0$, $x_k = x$ and $\langle x_{i-1}, \sigma_i, x_i \rangle \in \text{Edges}$ for all $1 \leq i \leq k$.

Algorithm 1 attempts to construct an L model of $\langle \Gamma, A \rangle$. The intended model extends A with disjoint trees rooted at state names occurring in A . The trees may be infinite. However, we represent such a semi-forest as a graph with global caching: if two unnamed states (i.e., states not occurring in A) in a tree or in different trees have the same label, then they should be merged. In other words, for every finite set X of formulas, the graph contains at most one state (a node)

Function Find(X)

```

1 if there exists  $z \in \Delta \setminus \Delta_0$  with  $\text{Label}(z) = X$  then return  $z$ ;
2 else add a new element  $z$  to  $\Delta$  with  $\text{Label}(z) := X$  and return  $z$ ;

```

Procedure ExtendLabel(z, X)

```

1 if  $X \subseteq \text{Label}(z)$  then return;
2 if  $z \in \Delta_0$  then  $\text{Label}(z) := \text{Label}(z) \cup \text{Sat}(X)$ ;
3 else // simulate changing label
4    $z_* := \text{Find}(\text{Label}(z) \cup \text{Sat}(X))$ ;
5   foreach  $y, \sigma, \varphi$  such that  $\text{Next}(y, \langle\sigma\rangle\varphi) = z$  do  $\text{Next}(y, \langle\sigma\rangle\varphi) := z_*$ ;

```

Function CheckPremise($x, A_1 \wedge \dots \wedge A_k$)

```

1 foreach  $1 \leq i \leq k$  do
2   if  $A_i = [\sigma]p$  then
3     if  $\text{Next}(x, \langle\sigma\rangle\top)$  is not defined then return false;
4     else if  $\text{Next}(x, \langle\sigma\rangle\top) = y$  and  $p \notin \text{Label}(y)$  then return false;
5     else if  $A_i = \langle\sigma\rangle p$  and  $\langle\mathbf{A}_\sigma\rangle p \notin \text{Label}(x)$  then return false;
6     else if  $A_i = p$  and  $p \notin \text{Label}(x)$  then return false;
7   return true;

```

$z \in \Delta \setminus \Delta_0$ such that $\text{Label}(z) = X$. The function $\text{Find}(X)$ returns such a state z if it exists, or creates such a state z otherwise. A tuple $\langle x, \sigma, y \rangle \in \text{Edges}$ represents an edge $\langle x, y \rangle$ with label σ of the graph. The notions of *predecessor*, *successor*, *ancestor* and *descendant* are defined as usual.

For each $x \in \Delta$, $\text{Label}(x)$ is a set of requirements to be “realized” at x . To realize such requirements of states, sometimes we have to extend their labels. Suppose we want to extend the label of $z \in \Delta$ with a set X of formulas. Consider the following cases:

- Case $z \in \Delta_0$ (i.e., z is a named state occurring in \mathcal{A}): as z is “fixed” by the ABox \mathcal{A} , we have no choice but to extend $\text{Label}(z)$ directly with $\text{Sat}(X)$.
- Case $z \notin \Delta_0$ and the requirements X are directly caused by z itself or its successors: if we directly extend the label of z (with $\text{Sat}(X)$) then z will possibly have the same label as another state not belonging to Δ_0 and global caching is not fulfilled. Hence, we “simulate” changing the label of z by using $z_* := \text{Find}(\text{Label}(z) \cup \text{Sat}(X))$ for playing the role of z . In particular, for each y, σ and φ such that $\text{Next}(y, \langle\sigma\rangle\varphi) = z$, we set $\text{Next}(y, \langle\sigma\rangle\varphi) := z_*$.

Extending the label of z for the above two cases is done by Procedure $\text{ExtendLabel}(z, X)$ given on page 7. The third case is considered below.

Suppose that $\text{Next}(x, \langle\sigma\rangle\varphi) = y$. Then, to realize the requirements at x , the label of y should be extended with $X = \text{Sat}(\text{Trans}(\text{Label}(x), \sigma))$. How can we realize such an extension? Recall that we intend to construct a forest-like model for $\langle\Gamma, \mathcal{A}\rangle$, but use global caching to guarantee termination. There may exist

Algorithm 1: checking satisfiability of a Horn knowledge base $\langle \Gamma, \mathcal{A} \rangle$ in L .

Input: L is $SL_{\Sigma, S}$ or $SL_{\Sigma+, S}$, where S is a regular semi-Thue system specified by a mapping \mathbf{A} , Γ is a finite set of Horn clauses, and \mathcal{A} is an ABox.

Output: *true* if $\langle \Gamma, \mathcal{A} \rangle$ is satisfiable in L , or *false* otherwise.

```

1 let  $\Delta_0$  be the set of all state names occurring in  $\mathcal{A}$ ;
2 if  $\Delta_0 = \emptyset$  then  $\Delta_0 := \{\tau\}$ ;
3 set  $\Delta := \Delta_0$  and set  $Next$  to the empty mapping;
4 foreach  $a \in \Delta_0$  do  $Label(a) := \text{Sat}(\{\varphi \mid \varphi(a) \in \mathcal{A}\} \cup \Gamma)$ ;
5 repeat
6   foreach  $\sigma(a, b) \in \mathcal{A}$  do  $Label(b) := Label(b) \cup \text{Sat}(\text{Trans}(Label(a), \sigma))$ ;
7   foreach  $x, \langle \sigma \rangle \varphi, y$  such that  $Next(x, \langle \sigma \rangle \varphi) = y$  and  $x$  is reachable
      from  $\Delta_0$  do
8      $y_* := \text{Find}(Label(y) \cup \text{Sat}(\text{Trans}(Label(x), \sigma)))$ ;
9      $Next(x, \langle \sigma \rangle \varphi) := y_*$ ;
10    foreach  $\langle x, \sigma, y \rangle \in Edges$  such that  $x$  is reachable from  $\Delta_0$  do
11       $\text{ExtendLabel}(x, \text{Trans}(Label(y), \bar{\sigma}))$ ;
12    foreach  $x \in \Delta$  reachable from  $\Delta_0$  and  $\langle \sigma \rangle \varphi \in Label(x)$  do
13      if  $Next(x, \langle \sigma \rangle \varphi)$  is not defined then
14         $Next(x, \langle \sigma \rangle \varphi) := \text{Find}(\text{Sat}(\{\varphi\} \cup \text{Trans}(Label(x), \sigma) \cup \Gamma))$ ;
15    foreach  $x \in \Delta$  reachable from  $\Delta_0$  and  $(\varphi \rightarrow \psi) \in Label(x)$  do
16      if  $\text{CheckPremise}(x, \varphi)$  then  $\text{ExtendLabel}(x, \{\psi\})$ ;
17    if there exist  $x \in \Delta$  and  $\{p, \neg p\} \subseteq Label(x)$  or  $\perp \in Label(x)$  then
18      return false;
19 until no changes occurred in the last iteration;
20 return true;
```

another $Next(x', \langle \sigma' \rangle \varphi') = y$ with $x' \neq x$. That is, we may use y as a successor for two different states x and x' , but the intention is to put x and x' into disjoint trees. If we directly modify the label of y to realize the requirements of x , such a modification may affect x' . The solution is to delete the edge $\langle x, \sigma, y \rangle$ and reconnect x to $y_* := \text{Find}(Label(y) \cup X)$ by setting $Next(x, \langle \sigma \rangle \varphi) := y_*$. The extension is formally realized by steps 7-9 of Algorithm 1.

Consider the other main steps of Algorithm 1:

- Step 6: If $\sigma(a, b) \in \mathcal{A}$ then we directly extend the label of b with $\text{Sat}(\text{Trans}(Label(a), \sigma))$.
- Steps 10-11: If $\langle x, \sigma, y \rangle \in Edges$ then we extend the label of x with $\text{Trans}(Label(y), \bar{\sigma})$ by using the procedure `ExtendLabel` discussed earlier. Notice the converse $\bar{\sigma}$.
- Steps 12-14: If $\langle \sigma \rangle \varphi \in Label(x)$ and $Next(x, \langle \sigma \rangle \varphi)$ is not defined yet then to realize the requirement $\langle \sigma \rangle \varphi$ at x we connect x via σ to a node with label $X = \text{Sat}(\{\varphi\} \cup \text{Trans}(Label(x), \sigma) \cup \Gamma)$ by setting $Next(x, \langle \sigma \rangle \varphi) := \text{Find}(X)$.

- Steps 15-16: If $(\varphi \rightarrow \psi) \in Label(x)$ and φ “holds” at x then we extend the label of x with $\{\psi\}$ by using the procedure `ExtendLabel` discussed earlier. Suppose $\varphi = A_1 \wedge \dots \wedge A_k$. How to check whether φ “holds” at x ? It “holds” at x if A_i “holds” at x for every $1 \leq i \leq k$. If A_i has the form p then it “holds” at x if $p \in Label(x)$. For the cases when A_i has the form $[\sigma]p$ or $\langle \sigma \rangle p$ we apply the following special techniques.
 - * The first technique, for the case $A_i = [\sigma]p$, was used earlier in other papers [1,12,14]: we just check whether $p \in Label(y)$ with $y = Next(x, \langle \sigma \rangle \top)$. The intuition is that, y is the “least σ -successor” of x , and if $p \in Label(y)$ then p occurs or will occur in all σ -successors of x .
 - * The second technique, for the case $A_i = \langle \sigma \rangle p$, is first proposed in the current paper. It is as follows. Whenever p appears in $Label(z)$ for some state z , by saturation we include in $Label(z)$ also $[A_{\bar{\sigma}}]\langle A_{\sigma} \rangle p$. Note that $p \rightarrow [A_{\bar{\sigma}}]\langle A_{\sigma} \rangle p$ is valid. Then, to check whether $A_i = \langle \sigma \rangle p$ “holds” at x , we just check whether $\langle A_{\sigma} \rangle p \in Label(x)$. (Semantically, $\langle A_{\sigma} \rangle p$ is equivalent to $\langle \sigma \rangle p$.) The reason for using this technique is due to the use of global caching (in order to guarantee termination).

We do global caching to represent a possibly infinite semi-forest by a finite graph possibly with cycles. As a side effect, direct checking “realization” of existential automaton-modal operators is not safe. Furthermore, we cannot allow universal modal operators “run” along such cycles. “Running” universal modal operators backward along an edge is safe, but “running” universal modal operators forward along an edge is done using a special technique, which may replace the edge by another one as in the steps 7-9 of Algorithm 1. Formally, checking whether φ “holds” at x is done by Function `CheckPremise`(x, φ) given on page 7.

Expansions by modifying the label of a state and/or setting the mapping $Next$ are done only for states that are reachable from Δ_0 . Note that, when a node z is simulated by z_* as in Procedure `ExtendLabel`, the node z becomes unreachable from Δ_0 . We do not delete such nodes z because they may be reused later.

When some $x \in \Delta$ has $Label(x)$ containing \perp or a pair p and $\neg p$, Algorithm 1 returns *false*, which means that the knowledge base $\langle \Gamma, \mathcal{A} \rangle$ is unsatisfiable in L . When the graph cannot be expanded anymore, the algorithm terminates in the normal mode with result *true*, which means $\langle \Gamma, \mathcal{A} \rangle$ is satisfiable in L .

Theorem 3 *Algorithm 1 correctly checks satisfiability of Horn knowledge bases in $SL_{\Sigma,S}$ and $SL_{\Sigma+,S}$ and has PTIME data complexity.*

See the long version [6] of this paper for a proof of this theorem.

Corollary 4 *The problem of checking satisfiability of Horn knowledge bases in $SL_{\Sigma,S}$ and $SL_{\Sigma+,S}$ has PTIME data complexity.*

4. Conclusions

With the ability to express consistency, positive introspection, negative introspection of knowledge and belief as well as knowledge sharing between agents

or groups of agents, serial regular grammar logics with converse can be used for reasoning about epistemic states of multiagent systems as well as for many other application areas.

In this paper we have developed the first algorithm with PTIME data complexity for checking satisfiability of a Horn knowledge base in a serial regular grammar logic with converse. By this we established PTIME data complexity of the considered satisfiability problem. Our method extends the method of Nguyen [1] by using a special technique for dealing with converse.

Acknowledgements

This work was supported by the Polish National Science Centre (NCN) under Grants No. 2011/01/B/ST6/02769 and 2011/01/B/ST6/02759.

References

- [1] L. Nguyen, “Constructing finite least Kripke models for positive logic programs in serial regular grammar logics,” *Logic Journal of the IGPL*, vol. 16, no. 2, pp. 175–193, 2008.
- [2] S. Demri and H. de Nivelle, “Deciding regular grammar logics with converse through first-order logic,” *Journal of Logic, Language and Inform.*, vol. 14, no. 3, pp. 289–329, 2005.
- [3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *Description Logic Handbook*. Cambridge University Press, 2002.
- [4] L. Nguyen, “Horn knowledge bases in regular description logics with PTime data complexity,” *Fundamenta Informaticae*, vol. 104, no. 4, pp. 349–384, 2010.
- [5] R. Goré and L. Nguyen, “A tableau system with automaton-labelled formulae for regular grammar logics,” in *Proceedings of TABLEAUX 2005* (B. Beckert, ed.), vol. 3702 of *LNAI*, pp. 138–152, Springer, 2005.
- [6] L. Nguyen and A. Szałas, “The long version of the current paper.” <http://www.mimuw.edu.pl/~nguyen/HSGRLC-long.pdf>, 2012.
- [7] M. Baldoni, L. Giordano, and A. Martelli, “A tableau for multimodal logics and some (un)decidability results,” in *TABLEAUX’1998, LNCS 1397:44-59*, 1998.
- [8] S. Demri, “The complexity of regularity in grammar logics and related modal logics,” *Journal of Logic and Computation*, vol. 11, no. 6, pp. 933–960, 2001.
- [9] B. Dunin-Kęplicz, L. Nguyen, and A. Szałas, “A framework for graded beliefs, goals and intentions,” *Fundamenta Informaticae*, vol. 100, no. 1–4, pp. 53–76, 2010.
- [10] B. Dunin-Kęplicz, L. Nguyen, and A. Szałas, “Converse-PDL with regular inclusion axioms: A framework for MAS logics,” *J. Applied Non-Classical Logics*, vol. 21, no. 1, pp. 61–91, 2011.
- [11] P. Doherty, B. Dunin-Kęplicz, and A. Szałas, “Dynamics of approximate information fusion,” in *Proc. RSEISP 2007* (M. Kryszkiewicz, J. Peters, H. Rybinski, and A. Skowron, eds.), no. 4585 in *LNAI*, pp. 668–677, Springer, 2007.
- [12] B. Dunin-Kęplicz, L. Nguyen, and A. Szałas, “Tractable approximate knowledge fusion using the Horn fragment of serial propositional dynamic logic,” *Int. J. Approx. Reasoning*, vol. 51, no. 3, pp. 346–362, 2010.
- [13] A. Mateescu and A. Salomaa, “Formal languages: an introduction and a synopsis,” in *Handbook of Formal Languages - Volume 1*, pp. 1–40, Springer, 1997.
- [14] L. Nguyen, “Constructing the least models for positive modal logic programs,” *Fundamenta Informaticae*, vol. 42, no. 1, pp. 29–60, 2000.

Cut-Free ExpTime Tableaux for Converse-PDL Extended with Regular Inclusion Axioms

Linh Anh NGUYEN ¹

*Institute of Informatics, University of Warsaw, Poland;
VNU University of Engineering and Technology, Hanoi, Vietnam*

Abstract. We develop a cut-free tableau calculus for the logic CPDL_{reg} , leading to the first cut-free EXPTIME (optimal) tableau decision procedure for CPDL_{reg} . This logic extends Converse-PDL with regular inclusion axioms characterized by finite automata. It is a logical formalism suitable for expressing complex properties of agents' cooperation in terms of beliefs, goals and intentions.

Keywords. CPDL, regular grammar logics, tableaux, global state caching

1. Introduction

In this paper we study automated reasoning in the modal logic CPDL_{reg} , which is a combination of CPDL (propositional dynamic logic with converse [1]) and REG^c (regular grammar logic with converse [2,3]). The logic CPDL is widely used in many areas, including program verification, theory of action and change, as well as knowledge representation. The logic REG^c can be used for modeling and reasoning about epistemic states of multi-agent systems and Web ontologies.

In [4,5] together with Dunin-Kęplicz and Szałas we showed that the combined logic CPDL_{reg} is a formalism suitable for expressing complex properties of agents' cooperation in terms of beliefs, goals and intentions. In particular, the multimodal formalism TEAMLOG [6,7] for specifying cooperating BDI agents and reasoning about teamwork as well as its graded version TEAMLOG^K [4] can be translated into CPDL_{reg} . The works [4,5] present a tableau calculus leading to an EXPETIME (optimal) tableau decision procedure for CPDL_{reg} , which can be used as a decision tool for multiagent formalisms.

The tableau calculus given in [4,5], however, uses analytic cuts and therefore is not efficient in practice. In this paper we improve that calculus by eliminating cuts to give the first cut-free EXPETIME (optimal) tableau decision procedure for CPDL_{reg} . Our calculus uses global state caching [8,9], which modifies global caching [10,11] for dealing with converse without using cuts. It also uses local caching for non-states of tableaux as well as a slightly different technique for dealing with converse modal operators.

¹Corresponding Author: Linh Anh Nguyen, Institute of Informatics, University of Warsaw, Banacha 2, 02-097, Warsaw, Poland; E-mail: nguyen@mimuw.edu.pl.

Cut rules are “or”-rules for guessing the “future”. They are usually used in a systematic way, which generates a lot of branches. Eliminating cuts is very important for efficient automated reasoning in CPDL_{reg}.

The rest of this paper is structured as follows. In Section 2 we give definitions for the logic CPDL_{reg}. In Section 3 we present our tableau calculus for CPDL_{reg}. We discuss an optimization technique in Section 4 and conclude in Section 5. Due to the lack of space, proofs of our results are presented only in the long version [12] of the current paper. An illustrative example and more explanations are also given in [12].

2. Preliminaries

Let Σ_+ be a finite set of symbols. For $\sigma \in \Sigma_+$, we use σ^- to denote a fresh symbol, called the *converse* of σ . Let $\Sigma_- = \{\sigma^- \mid \sigma \in \Sigma_+\}$ and $\Sigma = \Sigma_+ \cup \Sigma_-$. We call Σ an *alphabet with converse*. For $\rho = \sigma^- \in \Sigma_-$, define $\rho^- = \sigma$.

A *context-free semi-Thue system* S over Σ is a finite set of context-free production rules over alphabet Σ . (It is like a context-free grammar, but it has no designated start symbol and there is no distinction between terminal and non-terminal symbols.) We say that S is *symmetric* if, for every rule $\sigma \rightarrow \rho_1 \dots \rho_k$ of S , the rule $\sigma^- \rightarrow \rho_k^- \dots \rho_1^-$ also belongs to S . A context-free semi-Thue system S over Σ is called a *regular semi-Thue system* S over Σ if, for every $\sigma \in \Sigma$, the set of words derivable from σ using the system is a regular language over Σ .

Similarly to [2], we assume that any regular semi-Thue system S is given together with a mapping \mathbf{A} that associates each $\sigma \in \Sigma$ with a finite automaton \mathbf{A}_σ recognizing words derivable from σ using S . We assume that for $\sigma \in \Sigma$, the word σ is derivable from σ by such a system. We call \mathbf{A} the *mapping specifying the finite automata of S* .

A *finite automaton* A over alphabet Σ is a tuple $\langle \Sigma, Q, I, \delta, F \rangle$, where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation, and $F \subseteq Q$ is the set of accepting states. A *run* of A on a word $\rho_1 \dots \rho_k$ is a finite sequence of states q_0, q_1, \dots, q_k such that $q_0 \in I$ and $\delta(q_{i-1}, \rho_i, q_i)$ holds for every $1 \leq i \leq k$. It is an *accepting run* if $q_k \in F$. We say that A *accepts* a word w if there exists an accepting run of A on w . The set of words accepted by A is denoted by $\mathcal{L}(A)$.

We call elements of Σ_+ *atomic programs*, and call elements of Σ *simple programs*. We denote simple programs by letters like σ and ρ . We use letters like p, r, s to denote *propositions* (i.e., atomic formulas).

Formulas and *programs* in the *base language* of CPDL_{reg} are defined respectively by the following BNF grammar rules:

$$\begin{aligned}\varphi &::= \top \mid \perp \mid p \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \alpha \rangle \varphi \mid [\alpha] \varphi \\ \alpha &::= \sigma \mid \alpha ; \alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \alpha^- \mid \varphi ?\end{aligned}$$

We use letters like α, β to denote programs, and φ, ψ, ξ to denote formulas. Given binary relations R_1 and R_2 , by $R_1 \circ R_2$ we denote their relational composition.

A *Kripke model* is a pair $M = \langle \Delta^M, {}^M \rangle$, where Δ^M is a set of *states*, and M is an interpretation function that maps each proposition p to a subset p^M of Δ^M , and each atomic program $\sigma \in \Sigma_+$ to a binary relation σ^M on Δ^M . The interpretation function is extended to interpret complex formulas and complex programs as shown in Figure 1. Note that the definition of $(\sigma^-)^M$ is compatible with the assumption $(\sigma^-)^- = \sigma$.

$\top^M = \Delta^M$, $\perp^M = \emptyset$	$(\alpha; \beta)^M = \alpha^M \circ \beta^M$
$(\neg\varphi)^M = \Delta^M \setminus \varphi^M$, $(\varphi \rightarrow \psi)^M = (\neg\varphi \vee \psi)^M$	$(\alpha \cup \beta)^M = \alpha^M \cup \beta^M$
$(\varphi \wedge \psi)^M = \varphi^M \cap \psi^M$, $(\varphi \vee \psi)^M = \varphi^M \cup \psi^M$	$(\alpha^*)^M = (\alpha^M)^*$
$(\langle \alpha \rangle \varphi)^M = \{x \in \Delta^M \mid \exists y[\alpha^M(x, y) \wedge \varphi^M(y)]\}$	$(\alpha^-)^M = (\alpha^M)^{-1}$
$([\alpha] \varphi)^M = \{x \in \Delta^M \mid \forall y[\alpha^M(x, y) \rightarrow \varphi^M(y)]\}$	$(\varphi?)^M = \{(x, x) \mid \varphi^M(x)\}$

Figure 1. Interpretation of complex formulas and complex programs.

We write $M, w \models \varphi$ to denote $w \in \varphi^M$. For a set X of formulas, we write $M, w \models X$ to denote that $M, w \models \varphi$ for all $\varphi \in X$. If $M, w \models \varphi$ (respectively, $M, w \models X$), then we say that M satisfies φ (respectively, X) at w , and that φ (respectively, X) is satisfied at w in M . We say that M validates X (and X is valid in M) if $M, w \models X$ for all $w \in \Delta^M$.

Let S be a symmetric regular semi-Thue system over Σ . The CPDL_{reg} logic corresponding to S , denoted by CPDL(S), is characterized by the class of Kripke models M such that, for every rule $\sigma \rightarrow \rho_1 \dots \rho_k$ of S , we have that $\rho_1^M \circ \dots \circ \rho_k^M \subseteq \sigma^M$. Such a structure is called an *L-model*, for $L = \text{CPDL}(S)$.

Let L be a CPDL_{reg} logic and X, Γ be finite sets of formulas. We say that X is *L-satisfiable* w.r.t. the set Γ of global assumptions if there exists an *L*-model that validates Γ and satisfies X at some state.

A formula or a program is in *negation-and-converse normal form* (NCNF) if it does not use \rightarrow , uses \neg only immediately before propositions, and uses the converse program constructor \neg only for atomic programs.

Every formula φ (respectively, program α) can be transformed to a formula φ' (respectively, program α') in NCNF that is equivalent to φ (respectively, α) in the sense that for every Kripke model M , $\varphi^M = (\varphi')^M$ (respectively, $\alpha^M = (\alpha')^M$). For example, the NCNF of formula $\neg[((\sigma_1 \cup \sigma_2); \sigma_3^*; (\neg p)?)^-](q \vee \neg r)$ is $\langle p?; (\sigma_3^-)^*; (\sigma_1^- \cup \sigma_2^-) \rangle (\neg q \wedge r)$. In this paper we assume that formulas and programs are represented in NCNF and write $\bar{\varphi}$ to denote the NCNF of $\neg\varphi$.

The alphabet $\Sigma(\alpha)$ of a program α and the regular language $\mathcal{L}(\alpha)$ generated by α are specified as follows:

$\Sigma(\sigma) = \Sigma$	$\mathcal{L}(\sigma) = \{\sigma\}$
$\Sigma(\varphi?) = \{\varphi?\}$	$\mathcal{L}(\varphi?) = \{\varphi?\}$
$\Sigma(\beta; \gamma) = \Sigma(\beta) \cup \Sigma(\gamma)$	$\mathcal{L}(\beta; \gamma) = \mathcal{L}(\beta) \cdot \mathcal{L}(\gamma)$
$\Sigma(\beta \cup \gamma) = \Sigma(\beta) \cup \Sigma(\gamma)$	$\mathcal{L}(\beta \cup \gamma) = \mathcal{L}(\beta) \cup \mathcal{L}(\gamma)$
$\Sigma(\beta^*) = \Sigma(\beta)$	$\mathcal{L}(\beta^*) = (\mathcal{L}(\beta))^*$,

where for sets of words M and N , $M.N = \{\alpha\beta \mid \alpha \in M, \beta \in N\}$, $M^0 = \{\varepsilon\}$ (where ε is the empty word), $M^{n+1} = M.M^n$ for $n \geq 0$, and $M^* = \bigcup_{n \geq 0} M^n$.

We will use letters like ω to denote either a simple program from Σ or a test (of the form $\varphi?$). A word $\omega_1 \dots \omega_k \in \mathcal{L}(\alpha)$ can be treated as the program $(\omega_1; \dots; \omega_k)$, especially when it is interpreted in a Kripke model. As a finite automaton A over alphabet $\Sigma(\alpha)$ corresponds to a program (the regular expression recognizing the same language), it is interpreted in a Kripke model as follows:

$$A^M = \bigcup \{\gamma^M \mid \gamma \in \mathcal{L}(A)\}. \quad (1)$$

3. A Tableau Calculus for CPDL_{reg}

From now on, let S be a symmetric regular semi-Thue system over Σ , \mathbf{A} be the mapping specifying the finite automata of S , and L be the CPDL_{reg} logic corresponding to S . We will present a tableau calculus for checking L -satisfiability.

For each program α , let \mathbb{A}_α be a finite automaton recognizing the regular language $\mathcal{L}(\alpha)$. For each program $\alpha \notin \Sigma$, let \mathbf{A}_α be a finite automaton recognizing the language $\mathcal{L}(\alpha')$, where α' is obtained from α by substituting each $\sigma \in \Sigma$ not inside any test by a regular expression representing $\mathcal{L}(\mathbf{A}_\sigma)$.

The automaton \mathbb{A}_α can be constructed from α in polynomial time, and \mathbf{A}_α can be constructed in polynomial time in the length of α and the sizes of the automata $(\mathbf{A}_\sigma)_{\sigma \in \Sigma}$. Roughly speaking, \mathbf{A}_α can be obtained from \mathbb{A}_α by simultaneously substituting each transition (q_1, σ, q_2) by the automaton \mathbf{A}_σ .

From now on, let X and Γ be finite sets of formulas in NCNF of the base language. For the tableau calculus defined here for checking L -satisfiability of X w.r.t. the set Γ of global assumptions we extend the base language with the auxiliary modal operators \Box_σ , $[A, q]$ and $\langle A, q \rangle$, where $\sigma \in \Sigma$, A is either \mathbf{A}_α or \mathbb{A}_α for some program α occurring in X or Γ in the form $[\alpha]\varphi$ or $\langle \alpha \rangle \varphi$, and q is a state of A . Here, $[A, q]$ and $\langle A, q \rangle$ stand respectively for $[(A, q)]$ and $\langle (A, q) \rangle$, where (A, q) is the automaton that differs from A only in that q is its only initial state. We call $[A, q]$ (respectively, $\langle A, q \rangle$) a *universal* (respectively, *existential*) *automaton-modal operator*.

In the *extended language*, if φ is a formula, then $\Box_\sigma \varphi$, $[A, q]\varphi$ and $\langle A, q \rangle \varphi$ are also formulas. A formula $\Box_\sigma \varphi$ has the same semantics as $[\sigma]\varphi$, i.e. $(\Box_\sigma \varphi)^M = ([\sigma]\varphi)^M$. The semantics of formulas $[A, q]\varphi$ and $\langle A, q \rangle \varphi$ are defined as usual, treating (A, q) as a program with semantics specified by (1). Despite that $\Box_\sigma \varphi$ has the same semantics as $[\sigma]\varphi$, the operator \Box_σ behaves differently from $[\sigma]$ in our calculus. We will use the following convention:

- given a finite automaton A , we always assume that $A = \langle \Sigma_A, Q_A, I_A, \delta_A, F_A \rangle$,
- for $q \in Q_A$, we define $\delta_A(q) = \{(\omega, q') \mid (q, \omega, q') \in \delta_A\}$.

In what follows we define tableaux as rooted “and-or” graphs. Such a graph is a tuple $G = (V, E, v_r)$, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $v_r \in V$ is the root, and each node $v \in V$ has a number of attributes. If $(v, w) \in E$ then we call v a *predecessor* of w , and call w a *successor* of v . The set of all attributes of v is called the *contents* of v . Attributes of tableau nodes are:

- $Type(v) \in \{\text{state}, \text{non-state}\}$. If $Type(v) = \text{state}$ then we call v a *state*, else we call v a *non-state* node). A state is never directly connected to a state.
- $Status(v) \in \{\text{unexpanded}, \text{expanded}, \text{incomplete}, \text{unsat}, \text{sat}\}$.
- $Label(v)$ is a finite set of formulas, called the label of v .
- $RFmls(v)$ is a finite set of formulas, called the set of reduced formulas of v .
- $DFmls(v)$ is a finite set of formulas, called the set of disallowed formulas of v .
- $StatePred(v) \in V \cup \{\text{null}\}$ is called the state-predecessor of v . It is available only when $Type(v) = \text{non-state}$. If v is a non-state and G has no paths connecting a state to v then $StatePred(v) = \text{null}$. Otherwise, G has exactly one state u such that u is a state and $(u, v) \in E$.

that is connected to v via a path not containing any other states. In that case, $\text{StatePred}(v) = u$.

- $\text{CELabel}(v)$ is a formula of the form $\langle \sigma \rangle \varphi$. It is called the coming edge label of v and is available only when v is a successor of a state. Informally, if $\text{CELabel}(v) = \langle \sigma \rangle \varphi$ and u is the predecessor of v then $\langle \sigma \rangle \varphi \in \text{Label}(u)$ and this formula is realized at u by creating a transition to v (transitions are done only for states).
- $\text{ATPred}(v) \in V$ is called the after-transition-predecessor of v . It is available only when $\text{Type}(v) = \text{non-state}$. If v is a non-state and $v_0 = \text{StatePred}(v) \neq \text{null}$ then there is exactly one successor v_1 of v_0 such that every path connecting v_0 to v must go through v_1 , and we have that $\text{ATPred}(v) = v_1$. If v is a non-state and $\text{StatePred}(v) = \text{null}$ then $\text{ATPred}(v)$ is set to the root v_r .
- $\text{FmlSC}(v)$ is a formula called the formula suggested by converse. It is available only when $\text{Type}(v) = \text{state}$ and $\text{Status}(v) = \text{incomplete}$. In that case, it means that some node w with $\text{StatePred}(w) = v$ wants $\varphi \in \text{Label}(v) \cup \text{RFmls}(v)$ but it does not hold. The formula φ is expected to be present at v when, for $v_1 = \text{ATPred}(w)$, $\text{CELabel}(v_1)$ is of the form $\langle \sigma \rangle \psi$ and $\Box_{\sigma^-} \varphi \in \text{Label}(w)$.

We define

$$\begin{aligned}\text{AFmls}(v) &= \text{Label}(v) \cup \text{RFmls}(v) \\ \text{Kind}(v) &= \begin{cases} \text{and-node if } \text{Type}(v) = \text{state} \\ \text{or-node if } \text{Type}(v) = \text{non-state} \end{cases}\end{aligned}$$

$\text{BeforeFormingState}(v) = v$ has a successor which is a state

$$\text{AfterTrans}(v) = (\text{ATPred}(v) = v).$$

$\text{AFmls}(v)$ is called the available formulas of v . In an “and-or” graph, states play the role of “and”-nodes, while non-states play the role of “or”-nodes.

By the *local graph* of a state v we mean the subgraph of G consisting of all the paths starting from v and not containing any other states. Similarly, by the local graph of a non-state v we mean the subgraph of G consisting of all the path starting from v and not containing any states.

We apply global state caching in the sense that if v_1 and v_2 are different states then $\text{Label}(v_1) \neq \text{Label}(v_2)$ or $\text{RFmls}(v_1) \neq \text{RFmls}(v_2)$ or $\text{DFmls}(v_1) \neq \text{DFmls}(v_2)$. If v is a non-state such that $\text{AfterTrans}(v)$ then we also apply global caching for the local graph of v . That is, if w_1 and w_2 are different nodes of the local graph of v then $\text{Label}(w_1) \neq \text{Label}(w_2)$ or $\text{RFmls}(w_1) \neq \text{RFmls}(w_2)$ or $\text{DFmls}(w_1) \neq \text{DFmls}(w_2)$.

Our calculus \mathcal{CL} for the CPDL_{reg} logic L will be specified, amongst others, by a finite set of tableau rules, which are used to expand nodes of tableaux. A *tableau rule* is specified with the following information: the kind of the rule (an “and”-rule or an “or”-rule), the conditions for applicability of the rule (if any), the priority of the rule, the number of successors of a node resulting from applying the rule to it, and the way to compute their contents.

Tableau rules are usually written downwards, with a set of formulas above the line as the *premise*, which represents the label of the node to which the rule is applied, and a number of sets of formulas below the line as the (*possible*) *conclusions*, which represent the labels of the successor nodes resulting from the application of the rule. Possible conclusions of an “or”-rule are separated by |, while conclusions of an “and”-rule are

separated by $\&$. If a rule is a unary rule (i.e. a rule with only one possible conclusion) or an “and”-rule then its conclusions are “firm” and we ignore the word “possible”. The meaning of an “or”-rule is that if the premise is L -satisfiable w.r.t. Γ then some of the possible conclusions are also L -satisfiable w.r.t. Γ , while the meaning of an “and”-rule is that if the premise is L -satisfiable w.r.t. Γ then all of the conclusions are also L -satisfiable w.r.t. Γ (possibly in different states of the model under construction).

Such a representation gives only a part of the specification of the rules.

We use Y to denote a set of formulas, write Y, φ to denote $Y \cup \{\varphi\}$ and write Y, Γ to denote $Y \cup \Gamma$. Our *tableau calculus* \mathcal{CL} for the CPDL_{reg} logic L w.r.t. the set Γ of global assumptions consists of the rules which are partially specified in Table 1 together with two special rules (*forming-state*) and (*conv*), which will be explained later.

For any rule of \mathcal{CL} except (*forming-state*) and (*conv*), the distinguished formulas of the premise are called the *principal formulas* of the rule. The rules (*forming-state*) and (*conv*) have no principal formulas.

We assume that, for each rule of \mathcal{CL} described in Table 1, the principal formulas are not members of the set Y which appears in the premise of the rule.

The rule (*trans*) is the only “and”-rule and the only *transitional rule*. Instantiating this rule, for example, to the set $\{\langle \sigma \rangle p, \langle \sigma \rangle q, \Box_{\sigma} r\}$ as the premise and $\Gamma = \{s\}$ we get two conclusions: $\{p, r, s\}$ and $\{q, r, s\}$. Expanding a state v in a tableau by the rule (*trans*), each successor w_i of v is created due to a corresponding principal formula $\langle \sigma_i \rangle \varphi_i$ of the rule and we have $CELabel(w_i) = \langle \sigma_i \rangle \varphi_i$.

The other rules of \mathcal{CL} are “or”-rules, which are also called *static rules*. The intuition behind distinguishing between static/transitional is that static rules do not change the state of the model under construction, while each conclusion of the transitional rule forces a move to a new state. The transitional rule is used to expand states, while the static rules are used to expand non-states.

For any state w , every predecessor v of w is always a non-state. Such a node v is expanded and connected to w by the static rule (*forming-state*). The nodes v and w correspond to the same state of the Kripke model under construction. In other words, the rule (*forming-state*) “transforms” a non-state to a state. The idea is to separate non-states from states, which are globally cached. The rule (*forming-state*) guarantees that, if $BeforeFormingState(v)$ holds then v has exactly one successor, which is a state.

Expanding a non-state v of a tableau by a static rule $\rho \in \{(\wedge), (\vee), (aut_{\Box}), (aut_{\Diamond}), ([A]), ([A]_f), (\Box_?)\}$ which uses φ as the principal formula we put φ into the set $RFmls(w)$ of each successor w of v by setting $RFmls(w) := RFmls(v) \cup \{\varphi\}$. We use $RFmls(w)$ to disallow expanding w by static rules which use a formula from $RFmls(w)$ as the principal formula (i.e. to block reducing the formulas from $RFmls(w)$ twice). If a non-state v is expanded by a static rule $\rho \in \{(\langle A \rangle), (\langle A \rangle_f), (\Diamond?), (conv)\}$ and w is a successor of v then we set $RFmls(w) := RFmls(v)$. Thus, we do not use the attribute $RFmls$ to disallow the rules $(\langle A \rangle), (\langle A \rangle_f), (\Diamond?)$ (in order to be able to fulfill eventualities of the form $\langle A, q \rangle \varphi$). If v is expanded by the rule (*trans*) and w is a successor of v then we set $RFmls(w) := \emptyset$.

Let v_0, v_1, \dots, v_k be a path of a tableau such that $k \geq 1$, $Type(v_0) = \text{state}$ and $Type(v_i) = \text{non-state}$ for $1 \leq i \leq k$. Suppose that $CELabel(v_1) = \langle \sigma \rangle \psi$ and $\Box_{\sigma} \varphi \in Label(v_k)$. If v_0 corresponds to a state x of a Kripke model M , then all v_1, \dots, v_k correspond to a state y of M such that $(x, y) \in \sigma^M$. The formulas from $Label(v_k)$ are supposed to be satisfied at y in M , which causes φ satisfied at x in M . So, φ is expected

$(\wedge) \frac{Y, \varphi \wedge \psi}{Y, \varphi, \psi}$	$(\vee) \frac{Y, \varphi \vee \psi}{Y, \varphi \mid Y, \psi}$
$(aut_{\square}) \frac{Y, [\alpha]\varphi}{Y, [\mathbf{A}_{\alpha}, q_1]\varphi, \dots, [\mathbf{A}_{\alpha}, q_k]\varphi} \text{ if } I_{\mathbf{A}_{\alpha}} = \{q_1, \dots, q_k\}$	
$(aut_{\diamond}) \frac{Y, \langle \alpha \rangle \varphi}{Y, \langle \mathbb{A}_{\alpha}, q_1 \rangle \varphi \mid \dots \mid Y, \langle \mathbb{A}_{\alpha}, q_k \rangle \varphi} \text{ if } \begin{cases} \alpha \notin \Sigma, \alpha \text{ is not a test,} \\ \text{and } I_{\mathbb{A}_{\alpha}} = \{q_1, \dots, q_k\} \end{cases}$	
if $\delta_A(q) = \{(\omega_1, q_1), \dots, (\omega_k, q_k)\}$ and $q \notin F_A$:	
$([A]) \frac{Y, [A, q]\varphi}{Y, \square_{\omega_1}[A, q_1]\varphi, \dots, \square_{\omega_k}[A, q_k]\varphi}$	
$(\langle A \rangle) \frac{Y, \langle A, q \rangle \varphi}{Y, \langle \omega_1 \rangle \langle A, q_1 \rangle \varphi \mid \dots \mid Y, \langle \omega_k \rangle \langle A, q_k \rangle \varphi}$	
if $\delta_A(q) = \{(\omega_1, q_1), \dots, (\omega_k, q_k)\}$ and $q \in F_A$:	
$([A]_f) \frac{Y, [A, q]\varphi}{Y, \square_{\omega_1}[A, q_1]\varphi, \dots, \square_{\omega_k}[A, q_k]\varphi, \varphi}$	
$(\langle A \rangle_f) \frac{Y, \langle A, q \rangle \varphi}{Y, \langle \omega_1 \rangle \langle A, q_1 \rangle \varphi \mid \dots \mid Y, \langle \omega_k \rangle \langle A, q_k \rangle \varphi \mid Y, \varphi}$	
$(\square ?) \frac{Y, \square_{(\psi?)}\varphi}{Y, \psi \mid Y, \varphi}$	$(\diamond ?) \frac{Y, \langle \psi? \rangle \varphi}{Y, \psi, \varphi}$
$(trans) \frac{Y, \langle \sigma_1 \rangle \varphi_1, \dots, \langle \sigma_k \rangle \varphi_k}{Y_1, \varphi_1, \Gamma \& \dots \& Y_k, \varphi_k, \Gamma}$	
where $\begin{cases} k \geq 1, Y \text{ contains no formulas of the form } \langle \sigma \rangle \varphi, \\ \text{and } Y_i = \{\psi : \square_{\sigma_i} \psi \in Y\} \text{ for every } 1 \leq i \leq k \end{cases}$	

Table 1. Some rules of the tableau calculus \mathcal{CL} for a CPDL_{reg} logic L .

to belong to AFmls(v_0) (i.e. we should realize φ at v_0). What should be done in the case $\varphi \notin \text{AFmls}(v_0)$? We will not simply add φ to $\text{Label}(v_0)$, as v_k is only one of possibly many “or”-descendants of v_0 , and adding φ to $\text{Label}(v_0)$ may affect the other “or”-descendants of v_0 (which is not allowed). If $\varphi \in \text{DFmls}(v_0)$, which means φ is disallowed in v_0 , then $\text{Status}(v_k)$ becomes unsat, which intuitively means that the “combination” of v_0 and v_k is L -unsatisfiable w.r.t. Γ . In the other case, $\text{Status}(v_0)$ becomes incomplete, the predecessors of v_0 will be re-expanded by the rule (*conv*) either to have φ (by adding φ to the attribute Label) or to disallow φ (by adding φ to DFmls).

The priorities of the tableau rules are as follows: unary static rules except (*forming-state*): 5; non-unary static rules: 4; (*forming-state*): 3; (*trans*): 2; (*conv*): 1. Here are the conditions for applying a rule $\rho \neq (\text{conv})$ to a node v :

- the rule has $\text{Label}(v)$ as the premise,
- all the conditions accompanying with ρ in Table 1 are satisfied,

- if $\rho = (\text{trans})$ then $Type(v) = \text{state}$,
- if $\rho \neq (\text{trans})$ then $Type(v) = \text{non-state}$
- * if $\rho \in \{(\wedge), (\vee), (\text{aut}_{\square}), (\text{aut}_{\diamond}), ([A]), ([A]_f), (\square_?)\}$
then the principal formula of ρ does not belong to $RFmls(v)$,
- * no static rule with a higher priority is applicable to v .

Application of a tableau rule ρ to a node v is specified by procedure $\text{Apply}(\rho, v)$ given on page 10 of [12]. Auxiliary functions are defined on page 9 of [12]. Procedures used for updating and propagating statuses of nodes as well as the main function $\text{Tableau}(X, \Gamma)$ are defined on page 11 of [12]. This function returns a rooted “and-or” graph called a \mathcal{CL} -tableau for (X, Γ) .

See [12] for an example of “and-or” graph.

A marking of a \mathcal{CL} -tableau G is a subgraph G_m of G such that:

- the root of G is the root of G_m ,
- if v is a node of G_m and is an “or”-node of G then at least one edge (v, w) of G is an edge of G_m ,
- if v is a node of G_m and is an “and”-node of G then every edge (v, w) of G is an edge of G_m ,
- if (v, w) is an edge of G_m then v and w are nodes of G_m .

Let G be a \mathcal{CL} -tableau for (X, Γ) , G_m be a marking of G , v be a node of G_m , and $\langle A, q \rangle \varphi$ be a formula of $\text{Label}(v)$. A trace of $\langle A, q \rangle \varphi$ in G_m starting from v is a sequence $(v_0, \varphi_0), \dots, (v_k, \varphi_k)$ such that:

- $v_0 = v$ and $\varphi_0 = \langle A, q \rangle \varphi$,
- for every $1 \leq i \leq k$, (v_{i-1}, v_i) is an edge of G_m ,
- for every $1 \leq i \leq k$, φ_i is a formula of $\text{Label}(v_i)$ such that
 - * if φ_{i-1} is not a principal formula of the tableau rule expanding v_{i-1} then the rule must be a static rule and $\varphi_i = \varphi_{i-1}$,
 - * else if the rule is $(\langle A \rangle)$ or $(\langle A \rangle_f)$ then φ_{i-1} is of the form $\langle A, q' \rangle \varphi$ and φ_i is the formula obtained from φ_{i-1} ,
 - * else if the rule is $(\diamond_?)$ then φ_{i-1} is of the form $\langle \psi? \rangle \langle A, q' \rangle \varphi$ and $\varphi_i = \langle A, q' \rangle \varphi$,
 - * else the rule is (trans) , φ_{i-1} is of the form $\langle \sigma \rangle \langle A, q' \rangle \varphi$ and is the coming edge label of v_i , and $\varphi_i = \langle A, q' \rangle \varphi$.

A trace $(v_0, \varphi_0), \dots, (v_k, \varphi_k)$ of $\langle A, q \rangle \varphi$ in a marking G_m is called a \diamond -realization in G_m for $\langle A, q \rangle \varphi$ at v_0 if $\varphi_k = \varphi$.

A marking G_m of a \mathcal{CL} -tableau G is consistent if:

local consistency: G_m does not contain nodes with status unsat; and

global consistency: for every node v of G_m , every formula $\langle A, q \rangle \varphi$ of $\text{Label}(v)$ has a \diamond -realization (starting from v) in G_m .

Theorem 1 (Soundness and Completeness) Let S be a symmetric regular semi-Thue system over Σ , A be the mapping specifying the finite automata of S , and L be the CPDL_{reg} logic corresponding to S . Let X and Γ be finite sets of formulas in NCNF of the base language, and G be a \mathcal{CL} -tableau for (X, Γ) . Then X is L -satisfiable w.r.t. the set Γ of global assumptions iff G has a consistent marking.

4. Propagating Statuses On-the-Fly

Based on the given tableau calculus, in [12] we provide an EXPTIME tableau decision procedure for CPDL_{reg} , which can be implemented with various optimization techniques [13]. One of basic optimization techniques is checking and propagating statuses of nodes on-the-fly. Whenever a node receives status sat or unsat, we update the statuses of its predecessors. In this section we discuss a technique for checking global inconsistency (of nodes) on-the-fly.

Let $G = (V, E, v_r)$ be a \mathcal{CL} -tableau for (X, Γ) and G_m be a marking of G . The graph G_t of traces of G_m in G is defined as follows:

- nodes of G_t are pairs (v, φ) , where $v \in V$ and $\varphi \in \text{Label}(v)$,
- a pair $((v, \varphi), (w, \psi))$ is an edge of G_t if v is a node of G_m , φ is of the form $\langle A, q \rangle \xi$ or $\langle \omega \rangle \langle A, q \rangle \xi$, and the sequence $(v, \varphi), (w, \psi)$ is a subsequence of a trace in G_m .

A node (v, φ) of G_t is an *end node* if φ is a formula of the base language. A node of G_t is *productive* if there is a path connecting it to an end node. Note that $(v, \langle A, q \rangle \varphi)$ is a productive node of G_t iff the formula $\langle A, q \rangle \varphi$ has a \Diamond -realization (starting from v) in G_m .

During the construction of G , let G_m be the subgraph of G induced by the nodes with status different from unsat and incomplete, and let G_t be constructed as described above. (When G becomes a full “and-or” graph, G_m will be a marking of G and G_t will be the graph of traces of G_m in G , provided that propagation of sat and unsat has been done on-the-fly.) At any stage, for nodes (v, φ) of G_t , define $\text{SemiEndNodes}(v, \varphi)$ to be the smallest sets of nodes of G_t that satisfy the following conditions:

- if (v, φ) is an end node of G_t or $\text{Status}(v) \in \{\text{unexpanded}, \text{sat}\}$ then $\text{SemiEndNodes}(v, \varphi) = \{(v, \varphi)\}$,
- else $\text{SemiEndNodes}(v, \varphi) = \bigcup \{\text{SemiEndNodes}(w, \psi) \mid (w, \psi) \text{ is a successor of } (v, \varphi) \text{ in } G_t \text{ different from } (v, \varphi)\} \setminus \{(v, \varphi)\}$.

The attribute SemiEndNodes for nodes of G_t can be computed and updated on-the-fly. Observe that, during the construction of G and G_t , if $\text{SemiEndNodes}(v, \varphi) = \emptyset$ then (v, φ) is not and will never be a productive node of G_t and hence $\text{Status}(v)$ can be set to unsat (and propagation of unsat can be conducted). Furthermore, when G becomes a full “and-or” graph, if a node (v, φ) of G_t has $\text{SemiEndNodes}(v, \varphi) \neq \emptyset$ then (v, φ) is a productive node of G_t .

Observe that the status of a node v of G depends only on the subgraph of G rooted at v . When that subgraph has been fully expanded and unsat could not be detected for v , we can set the status of v to sat and propagate it to the predecessors of v (in an appropriate way). The depth-first expansion strategy co-operates well with this technique.

5. Conclusions

We have developed a cut-free tableau calculus for the logic CPDL_{reg} , leading to the first cut-free EXPTIME (optimal) tableau decision procedure for checking satisfiability in CPDL_{reg} .

In comparison with the tableau calculus given in our joint papers [4,5] for CPDL_{reg} , the tableau calculus given in the current paper is essentially better since it is cut-free.

When restricted to REG^c , our tableau decision procedure is much better than the one proposed by us and Szałas in [3] for REG^c , as it does not use cuts and we do not have to check the global consistency property (this is an effect of cut elimination).

Apart from global caching of states as in [8,9], we allow to cache also non-states. We apply global caching for nodes in the local graphs of non-states v satisfying $\text{AfterTrans}(v)$. Such local graphs in [8,9] are trees. Furthermore, we propose not to delay solving incompatibility w.r.t. converse as long as in [8,9]. One can solve incompatibility w.r.t. converse as soon as possible as done in this paper. By giving a higher priority to the current branch even when it involves converse we can further favor depth-first search.

Acknowledgments

This work was supported by the Polish National Science Centre (NCN) under Grants No. 2011/01/B/ST6/02759 and 2011/01/B/ST6/02769.

References

- [1] D. Harel, D. Kozen, and J. Tiuryn, *Dynamic Logic*. MIT Press, 2000.
- [2] S. Demri and H. de Nivelle, “Deciding regular grammar logics with converse through first-order logic,” *Journal of Logic, Language and Inform.*, vol. 14, no. 3, pp. 289–329, 2005.
- [3] L. Nguyen and A. Szałas, “A tableau calculus for regular grammar logics with converse,” in *Proceedings of CADE-22* (R. Schmidt, ed.), vol. 5663 of *LNAI*, pp. 421–436, Springer-Verlag, 2009.
- [4] B. Dunin-Kęplicz, L. Nguyen, and A. Szałas, “A framework for graded beliefs, goals and intentions,” *Fundam. Inform.*, vol. 100, no. 1-4, pp. 53–76, 2010.
- [5] B. Dunin-Kęplicz, L. Nguyen, and A. Szałas, “Converse-PDL with regular inclusion axioms: a framework for MAS logics,” *Journal of Applied Non-Classical Logics*, vol. 21, no. 1, pp. 61–91, 2011.
- [6] B. Dunin-Kęplicz and R. Verbrugge, “Collective intentions,” *Fundam. Inform.*, vol. 51, no. 3, pp. 271–295, 2002.
- [7] M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz, “Complexity issues in multiagent logics,” *Fundam. Inform.*, vol. 75, no. 1-4, pp. 239–262, 2007.
- [8] R. Goré and F. Widmann, “Sound global state caching for ALC with inverse roles,” in *Proceedings of TABLEAUX 2009* (M. Giese and A. Waaler, eds.), vol. 5607 of *LNCS*, pp. 205–219, Springer, 2009.
- [9] R. Goré and F. Widmann, “Optimal and cut-free tableaux for propositional dynamic logic with converse,” in *Proceedings of IJCAR 2010* (J. Giesl and R. Hähnle, eds.), vol. 6173 of *LNCS*, pp. 225–239, Springer, 2010.
- [10] V. Pratt, “A near-optimal method for reasoning about action,” *J. Comp. Syst. Sci.*, vol. 20, no. 2, pp. 231–254, 1980.
- [11] R. Goré and L. Nguyen, “A tableau system with automaton-labelled formulae for regular grammar logics,” in *Proceedings of TABLEAUX 2005*, *LNAI 3702* (B. Beckert, ed.), pp. 138–152, Springer-Verlag, 2005.
- [12] L. Nguyen, “The long version of the current paper.” arXiv:1104.0405.
- [13] L. Nguyen, “An efficient tableau prover using global caching for the description logic ALC,” *Fundamenta Informaticae*, vol. 93, no. 1-3, pp. 273–288, 2009.

Abstract Test Suite Specification for ACL Communicating Agents

Marina BAGIĆ BABAC¹ and Dragan JEVTIĆ

Faculty of Electrical Engineering and Computing, Zagreb, Croatia

Abstract. Although accepted as a standard language for writing test specifications, TTCN-3 has not yet been used for an agent system testing. As TTCN-3 is best suited for black-box conformance testing of communicating systems, agents can be tested for conformance with their standard protocols. Therefore, we provide the abstract test suite for communicating agents exchanging ACL (Agent Communication Language) messages in order to achieve FIPA Request interaction protocol.

Keywords. TTCN-3, Coloured Petri nets, Model-based Testing

1. Introduction

TTCN-3 is a test specification language that applies to a variety of application domains and types of testing. As it provides all the constructs and features necessary for black box testing [1], in this paper we have proposed a framework for an agent system testing with TTCN-3. Specifically, we have tested agent system compliance with the standard FIPA Request Interaction Protocol [2].

We have used the strength of TTCN-3 [3] to build abstract test suite for communicating agents exchanging ACL (Agent Communication Language) messages [4]. While remaining abstract, our test suite does not depend on any particular agent platform implementation, therefore can be multiply reused.

Developing our framework we have first specified and verified a Request Interaction Protocol with Coloured Petri nets (CPNs), and then using the results of these, we have generated the agent abstract test suite in TTCN-3. The idea of using CPN model as a resource for creating TTCN-3 test suite has been recently published in [5], but it lacks any details on transformation between the CPN and TTCN-3.

There has been some work by others on testing agent systems in recent years. They have either focused on conformance testing for the properties of abstract BDI-agents [6], or unit testing for the plan based agent systems [7,8], or performed black box testing of the system [9] but not with TTCN-3.

The contribution of this paper also relates to the model-based testing [10], as a systematic and automated test case generation technique, being successfully applied to verify industrial-scale systems and is supported by commercial tools [11].

¹Corresponding Author: Marina Bagić Babac, University of Zagreb, Faculty of Electrical Engineering and Computing, HR-10000 Unska 3, Zagreb, Croatia; E-mail: marina.bagic@fer.hr

The interest in model-based testing is increasing since it promises early testing activities, hopefully early fault detection. However, scalability is still an open issue for large systems as in practice there are limits to the amount of testing that can be performed in industrial context [12].

The outline of this paper is as follows; in the Section 2 we have specified and verified the Request Interaction Protocol with CPN Tools [13]. In the Section 3 we have provided a TTCN-3 abstract framework for agent system testing. We have specified data types and templates, and test cases providing the testing facility. Loong Testing tool [14] have been used for testing simulation. Section 4 concludes the paper.

2. Specification of Communicating Agents with CP-nets

As an example of an agent system in this section we specify and verify the FIPA Request Interaction Protocol (FRIP) which allows one agent to request another to perform some action. The Participant agent processes the request and makes a decision whether to accept or refuse the request [2].

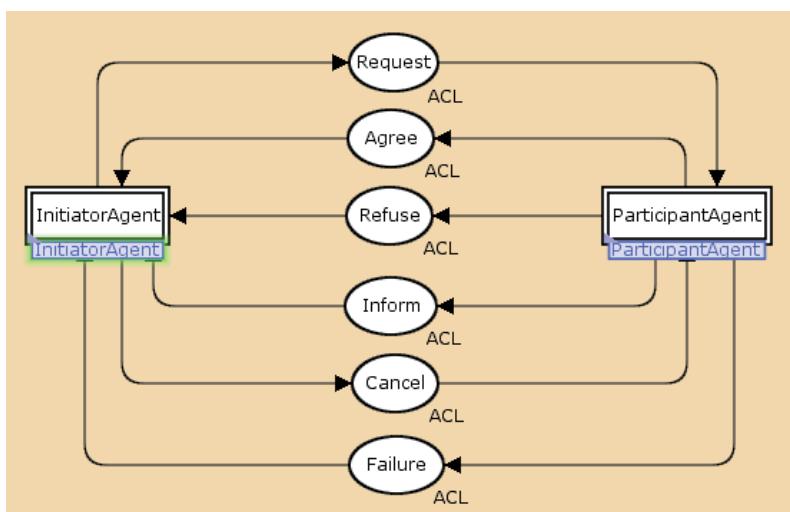
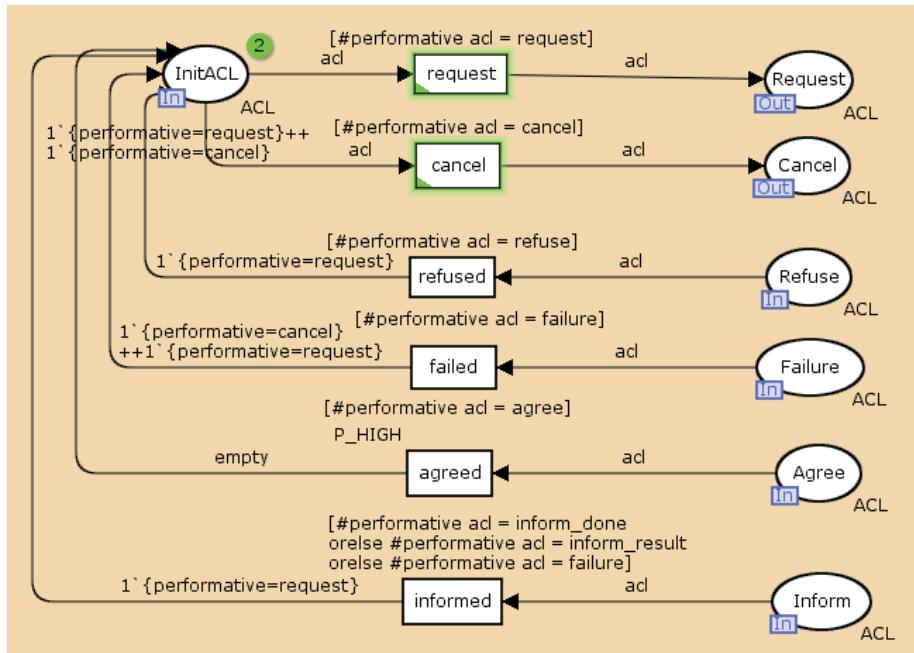
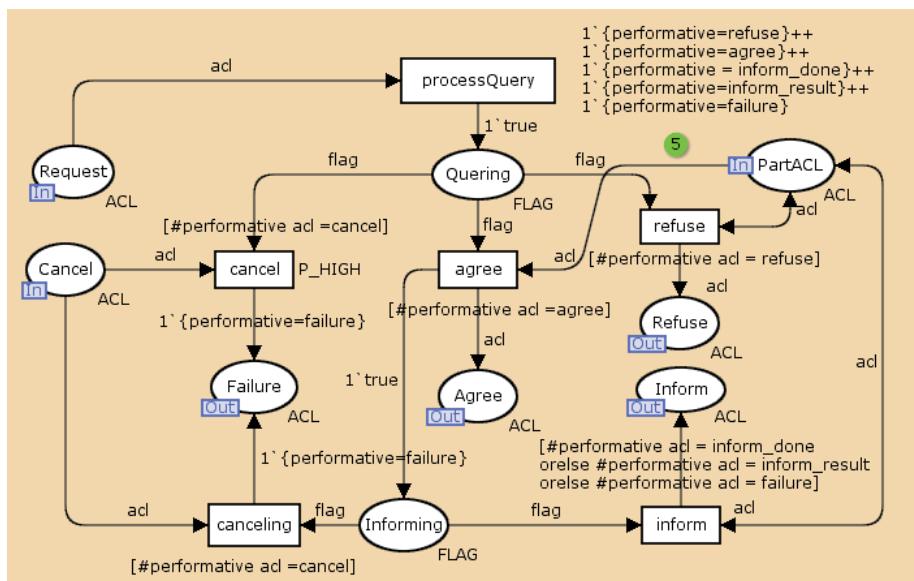


Figure 1. CPN Model for the Request Interaction Protocol

Once the request has been agreed upon, then the Participant must communicate either a failure if it fails in its attempt to fill the request, an inform-done if it completes the request and only indicates that it is done, or, an inform-result if it wishes to indicate both that it is done and notify the initiator of the results. Also, at any point the Initiator agent may cancel the interaction protocol.

We have used Coloured Petri nets [15], [16] for the specification of agents and CPN Tools [13] for the simulation and verification of their behaviours. Figure 1 shows an upper abstraction overview of agents, and Figure 2 and 3 specify the agents behaviours conforming to FIPA Request Interaction Protocol.

**Figure 2.** The Initiator Agent Specification**Figure 3.** The Participant Agent Specification

Verification results are based on generated state space tree with 18 nodes and 32 arcs between them. There are no dead markings and all markings are home ones, which means that the model never ends in a deadlock and never stays blocked forever waiting for the resources. Also, there are no dead transition instances and all of them are live. Best upper integer bounds of all places in the CPN model are limited implying that there is no endless accumulation of tokens in them. Due to fairness properties, there are two impartial, four fair and four just transition instances. There is also one transition with no fairness as we have given the "cancel" transition a high priority to stop both Initiator and Participant activities when necessary.

Having such formally defined and verified agents, we can use their CPN model for both TTCN-3 test cases generation as well as an agent system under test for testing with TTCN-3.

3. Specification of an Agent Abstract Test Suite

The next step after the specification and verification of the FRIP is creating TTCN-3 abstract test suite. A TTCN-3 test suite is a collection of test cases together with all the declarations and components needed for the test. The top-level unit of a TTCN-3 test suite is a module, which consists of a declarations part and a control part. The declarations part of a module contains definitions, while the control part specifies the order of the test cases invocation [3].

Figure 4 shows the test suite overview for Agent SUT (System Under Test) with concurrent test cases execution. There are multiple test components in the system, the Main Test Component (MTC), which is a mandatory component of each test suite, and a few Parallel test Components (PTCs) which are made by the MTC during the test execution [17].

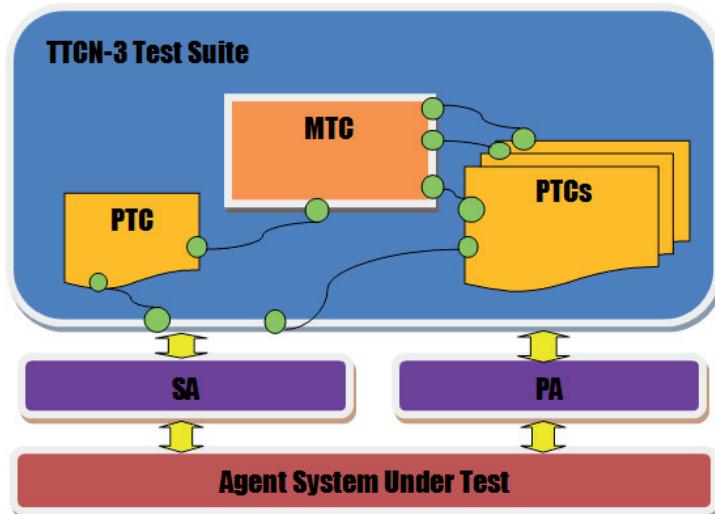


Figure 4. Test configuration for testing Agent SUT

Table 1. ACL Message Specification

```

type record ACLmessage {
    Performative performative,
    Sender sendAgent,
    Receiver receiveAgent,
    Content content,
    Reply_to reply_to optional,
    Language lang optional,
    Encoding encoding optional,
    Ontology ontology optional,
    Protocol protocol optional,
    Conversation_id conversation_id optional,
    Reply_with reply_with optional,
    In_reply_to in_reply_to optional,
    Reply_by reply_by optional
};
```

They are connected to each other and to Agent SUT via SA (System Adaptor) and PA (Platform Adaptor) [18] through the mechanism of communication ports. Green circles connected with curves in the figure represent ports for ACL message exchange among testing components.

In order to specify agent test suite we need to provide definitions for data types and templates to be exchanged during agent communications, port and component types, and the functions to be performed in the test cases executions.

3.1. Specification of Data, Port and Component Types

The initiating point of the abstract test suite specification is the choice of data types used in the test system. We must specify the protocol frame formats that need to be mapped to type definitions that are best represented in abstract form for test suite readability and maintainability reasons [19].

For effective communication between the agents in FRIP we use FIPA ACL [4] messages which contain a set of message parameters. Precisely which parameters are needed for effective agent communication will vary according to the situation; the only parameter that is mandatory in all ACL messages is the performative, although it is expected that most ACL messages will also contain sender, receiver and content parameters. In Table 1 we have specified an ACLmessage type using the record type while the parameters are specified using basic data types [20].

Once the message is sent from one agent to another, the sender agent continues its own actions in its environment with other agents and is not blocked waiting for the answer from the receiver agent. The communicating agents interact as equals in the system so their communication model is message-based (and not procedure-based) [21], and we define type ports accordingly. Ports are translated from our CPN model as I/O (Input/Output) places between the modules are called ports in CPN. The same concept holds for the test components.

Table 2. Port Types Specification

```
type port OutgoingACL message {
    out ACLmessage;
}
type port IncomingACL message {
    in ACLmessage;
}
```

Table 3. Agent Component Type Specification

```
type component AgentType {
    port IncomingACL incomingACL;
    port OutgoingACL outgoingACL;
}
```

Test components need to be defined with their interfaces (ports) towards the Agent SUT. In Table 2 there are two types of ports, one for the incoming messages and the other for the outgoing ones.

Type components are built upon these definitions of port types. The AgentType component type (Table 3) serves for both the initiator and participant agent testing. However, when multiple agents involved in the communication, AgentType component is extended with additional ports where port number depends on the number of initiator agents. Therefore, it is changeable and depends on the agent platform and its parameters for a runtime execution of an agent system.

3.2. Test Cases Generation

Modeling test cases includes specifying testing behaviour to be performed when the control part of an abstract test suite is invoked. We have derived these also from the CPN specifications in Figure 2 and 3 as they define an agent reactions to stimuli from another agent.

Function initiator() in Table 4 shows the initiator agent behaviour which is embedded within AgentType component. It first sends the "Request" ACL message, and then waits for the reply, but since this is not a blocking statement, the initiator agent could have been doing another task while waiting for the reply. After a while, it can either receive an "Agree" or a "Refuse" ACL message from the Participant agent. In both cases the verdict is set to pass as this is the expected behaviour of the participant agent. If the participant agent agrees to process the received request, it further sends another ACL message indicating the result of its work. It is the inform ("Inform_done" or "Inform_result") ACL message due to successfully fulfilled assignment, or the failure ACL message if it fails to fulfill the task.

An example of a test case is the one involving one initiator and one participant agent (Table 5). It is embedded within MTC component, so at system startup it creates both agents and connects their corresponding ports, and then invokes initiator() and participant() functions, respectively. Function participant() corresponds to the above initiator agent's requests.

Table 4. Function initiator() running on AgentType component

```

function initiator() runs on AgentType {
    outgoingACL.send(Request);
    alt {
        [] incomingACL.receive(Agree) {
            alt {
                [] incomingACL.receive(Inform) {
                    setverdict(pass);
                }
                [] incomingACL.receive(Failure) {
                    setverdict(pass);
                }
            }
        }
        [] incomingACL.receive(Refuse) {
            setverdict(pass)
        }
        [] incomingACL.receive {
            setverdict(fail)
        }
    }
}

```

Table 5. Test case specification

```
testcase testAgent () runs on EmptyComponentType {
    var AgentType InitiatorAgent;
    var AgentType ParticipantAgent;
    InitiatorAgent := AgentType.create;
    ParticipantAgent := AgentType.create;
    connect(InitiatorAgent:outgoingACL,
            ParticipantAgent:incomingACL);
    connect(InitiatorAgent:incomingACL,
            ParticipantAgent:outgoingACL);
    InitiatorAgent.start(initiator());
    ParticipantAgent.start(participant());
    timer t; t.start(6.0); t.timeout;
    InitiatorAgent.stop;
    ParticipantAgent.stop;
}
```

These functions contain sending and receiving message commands based on ACL messages from Table 1. Although agents communicate asynchronously, we have also included a timer control to prevent endless waiting for the response.

We have done the testing simulation using Loong Testing Tool [14] which provides the TTCN-3 compiler and an execution environment for an abstract test suite. Figure 5 shows a calling process trace indicating the order of an agent actions.

Calling Process							
Time	Type	Action	Module	Component	Object	Line No.	Component ID
Thu Nov 15 13:45:39 2012	Function	Enter into	FIPArequest		initiator	87	4
Thu Nov 15 13:45:39 2012	Function	Exit	FIPArequest		participant	107	5
Thu Nov 15 13:45:39 2012	Function	Exit	FIPArequest		initiator	93	4
Thu Nov 15 13:45:45 2012	Testcase	Exit	FIPArequest	MTC	testAgent	128	3
Thu Nov 15 13:45:45 2012		Exit	FIPArequest	control	control	128	1

Figure 5. Calling process trace with one initiator and one participant agent

The order suits the expected trace. Also, a result view pane (Figure 6) indicates the obtained pass verdicts by testing execution.

Result View Pane			
Seq. No.	Testcase	Module	Result
1	testAgent	FIPArequest	pass
2	testAgent	FIPArequest	pass

Figure 6. Obtained verdict with one initiator and one participant agent

Figure 7 shows the templates we have been using for ACL communication which are instantiated from the ACLMessage type (Table 1). It indicates that the templates fields are well suited for the expected agent communication. For each message a dedicated port is also outlined in the figure.

CODEC Information		Template	Matching Result
Port	Value		
outgoingACL	{performative := request,sendAgent := initiator1,receiveAgent := participant1,content := pay...}	{performative := request,sendAgent := initiator1,receiveA...	Matched
incomingACL	{performative := request,sendAgent := initiator1,receiveAgent := participant1,content := pa...	{performative := agree,sendAgent := participant1,receiveA...	Matched
outgoingACL	{performative := agree,sendAgent := participant1,receiveAgent := initiator1,content := pay...}	{performative := inform,sendAgent := participant1,receiveA...	Matched
incomingACL	{performative := agree,sendAgent := participant1,receiveAgent := initiator1,content := pay...}	{performative := inform,sendAgent := participant1,receiveA...	Matched
outgoingACL	{performative := inform,sendAgent := participant1,receiveAgent := initiator1,content := pay...}	{performative := inform,sendAgent := participant1,receiveA...	Matched
incomingACL	{performative := inform,sendAgent := participant1,receiveAgent := initiator1,content := pay...}	{performative := inform,sendAgent := participant1,receiveA...	Matched

Figure 7. Codec information with one initiator and one participant agent

Calling Process							
Time	Type	Action	Module	Component	Object	Line No.	Component ID
Thu Nov 15 17:43:16 2012		Enter into	FIPArequest	control	control	392	1
Thu Nov 15 17:43:16 2012	Testcase	Enter into	FIPArequest	MTC	testAgent	342	3
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		initiator	230	5
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		initiator2	244	6
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		initiator3	258	7
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		initiator4	272	8
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		participant	300	4
Thu Nov 15 17:43:16 2012	Function	Enter into	FIPArequest		initiator5	286	9
Thu Nov 15 17:43:17 2012	Testcase	Exit	FIPArequest	MTC	testAgent	384	3
Thu Nov 15 17:43:17 2012		Exit	FIPArequest	control	control	384	1

Figure 8. Calling process with five initiators and one participant agent

Another simulation is testing the behaviour of multiple agents in an agent system. Therefore, we have specified the system with five initiator agents communicating with one participant agent concurrently.

Figure 8 shows a calling process trace with the sequence of calling entities. Simulation testing has also proved that the agents have conformed to the specified FIPA Request Interaction Protocol.

4. Conclusion

This paper presents a black-box approach to testing an agent system using TTCN-3. TTCN-3 is a purely testing language at the high level of abstraction which provides us with test cases. Besides TTCN-3 we have also used Coloured Petri nets (CPN) for modeling, simulation and state space analysis of FIPA Request Interaction Protocol, which has served as a tutorial example for our approach. CPN model has served as a resource for extracting information on an agent behaviours and we have generated test cases according to CPN modules, places, transitions and colour sets. When coupled together CPN and TTCN-3 enable us to create tests which are applicable to both the specification model and implementation of an agent system. The transformation between the two languages has been done manually in this paper, so it remains a challenge to automatize some of the process for the future work.

Acknowledgement

This work was carried out within research project 036-0362027-1640 "Knowledge-based network and service management", supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

- [1] C. Willcock, S. Tobies, T. Deiss, S. Keil, F. Engler, S. Schulz, *An Introduction to TTCN-3*. John Wiley & Sons Ltd, 2005.
- [2] Foundation for Intelligent Physical Agents, *FIPA Request Interaction Protocol Specification*, SC00026H, FIPA TC Communication, 2002.
- [3] N. Lalani, *Validation of Internet Applications*. Fachhochschule Wiesbaden, Karlstad University, University of Applied Sciences, Sweden, Master Thesis, 2005.
- [4] Foundation for Intelligent Physical Agents, *FIPA ACL Message Structure Specification*, SC00061G, FIPA TC Communication, 2002.
- [5] T. Schnattinger, A. Pietschker, Using Colored Petri Nets for System Specifications and as a System Under Test Prototype, *TTCN-3 User Conference 2011*, 2011.
- [6] M. Zheng, V.S. Alagar, Conformance Testing of BDI Properties in Agent-based Software Systems, APSEC 05: *Proceedings of the 12th Asia-Pacific Software Engineering Conference APSEC05*, Washington. IEEE Computer Society, 457-464., 2005.
- [7] Z. Zhang, J. Thangarajah, L. Padgham, Automated testing for intelligent agent systems, AOSE'10 *Proceedings of the 11th international conference on Agent-oriented software engineering*, Springer-Verlag Berlin, Heidelberg, 66–79., 2010.
- [8] Z. Zhang, J. Thangarajah, L. Padgham, Automated Testing for Intelligent Agent Systems. AOSE 2009 *Proceedings of the 10th international conference on Agent-oriented software engineering*, 66–79., 2009.
- [9] H.-S.Seo, T. Araragi, T., Y.R. Kwon, Modeling and Testing Agent Systems Based on Statecharts, **3236** (2004), 308-321.
- [10] E. Bringmann, A. Kramer, Model-Based Testing of Automotive Systems, *2008 International Conference on Software Testing, Verification, and Validation*, (ICST), ISBN 978-0-7695-3127-4, 485-493, 2008.

- [11] H. Hemmati, A. Arcuri, L. Briand, *Achieving Scalable Model-Based Testing Through Test Case Diversity*, Technical report, Simula Research Laboratory, 2010.
- [12] M. Shafique, Y. Labiche, *A Systematic Review of Model Based Testing Tool Support*, Carleton University, Technical Report, May 2010.
- [13] CPN Tools, <http://cpn-tools.org/>, February 2013.
- [14] Loong Testing, <http://ttcn.usc.edu.cn/MainPageEn.html>, February 2013.
- [15] K. Jensen, L.M. Kristensen, *Coloured Petri Nets*, Springer-Verlag Berlin Heidelberg, 2009.
- [16] L.M. Kristensen, S. Christensen, K. Jensen, The practitioner's guide to coloured Petri nets, *International Journal on Software Tools for Technology Transfer*, **2** (1998), 98-132.
- [17] ETSI, Methods for Testing and Specification (MTS), The Testing and Test Control Notation version 3. *Part 1: TTCN-3 Core Language*, ETSI ES 201 873-1, V4.4.1, 2012.
- [18] ETSI, Methods for Testing and Specification (MTS), The Testing and Test Control Notation version 3. *Part 6: TTCN-3 Control Interface (TCI)*, ETSI ES 201 873-6 V4.4.1, 2012.
- [19] OpenTTCN Oy - OpenTTCN Ltd, OpenTTCN DocZone, *TTCN-3 language reference*, http://wiki.openttcn.com/media/index.php/OpenTTCN/Language_reference, February 2013.
- [20] W. Shaofeng, L. Fuliang, Y. Longguo, Z. Yanwei, Effective development and practice of automatic testing based on ttcn3, Presentation at *TTCN-3 User Conference 2010*, 2010.
- [21] M. Ebner, *An introduction to TTCN-3 version 3*. ITU-T Study Group 17, Geneva, 5-14th Oct 2005.
- [22] B. Stepien, TTCN-3 in a Nutshell. University of Ottawa, online tutorial, http://www.site.uottawa.ca/bernard/ttcn3_in_a_nutshell.html, February 2013.
- [23] TTCN-3 home page, <http://www.ttcn-3.org/>, February 2013.
- [24] A. Rennoch, C. Desroches, T. Vassiliou-Gioles, I. Schieferdecker, *TTCN-3 Reference Card*, <http://www.blukaktus.com/card.html>, online edition 4.4.1, April 2012.

Perceiving Speech Acts under Incomplete and Inconsistent Information¹

Barbara DUNIN-KĘPLICZ ^{a,2}, Alina STRACHOCKA ^b, Andrzej SZAŁAS ^c and Rineke VERBRUGGE ^d

^a Polish Academy of Sciences and Institute of Informatics, Warsaw University, Poland

^b Institute of Informatics, Warsaw University, Poland

^c Institute of Informatics, Warsaw University, Poland and Dept. of Computer and Information Science, Linköping University, Sweden

^d Institute of Artificial Intelligence, University of Groningen, The Netherlands

Abstract This paper discusses an implementation of four speech acts: *assert*, *concede*, *request* and *challenge* in a paraconsistent framework. A natural four-valued model of interaction yields multiple new cognitive situations. They are analyzed in the context of *communicative relations*, which partially replace the concept of trust. These assumptions naturally lead to six types of situations, which often require performing conflict resolution and belief revision.

The particular choice of a rule-based, DATALOG ^{“”}-like query language 4QL as a four-valued implementation framework ensures that, in contrast to the standard two-valued approaches, tractability of the model is achieved.

Keywords. paraconsistent modeling, communication, speech acts, tractable models, four-valued logic, conflict resolution

1. A Four-valued Formalism in Modeling Speech Acts

The development of multiagent systems (MAS) demands a precise modeling of the environment. An adequate knowledge representation method should be selected on an application-specific basis. When confining to logic-based approaches and formalisms, traditionally two-valued logics prevail. They fail, however, to express in a natural manner richer modeling aspects when some values or properties are simply unknown, or when the available information is inconsistent. A natural remedy for such situations is introducing four logical values. This work aligns with a whole line of research concerning logical modeling, reasoning and communicating about the surrounding reality, under the assumption that we deal with four types of situations, encoded in the four logical values: (i) fact *a* holds, (ii) fact *a* does not hold, (iii) it is not known whether *a* holds, (iv) information about *a* is inconsistent.

This paper opens our research program on paraconsistent modeling of communication in the four-valued framework. If argumentation-based dialogues are considered as communicative games between two or more agents, speech acts can be viewed as their

¹ Supported by the Polish National Science Centre grant 2011/01/B/ST6/02769.

² Corresponding Author: Barbara Dunin-Kęplicz, ICS, Polish Academy of Sciences and Institute of Informatics, Warsaw University, Poland, Email: keplicz@mimuw.edu.pl

building blocks. They compose complex dialogues, such as persuasion, deliberation, information seeking, negotiation or inquiry and form the underlying reactive layer of communication (see [1–7] for investigations in multi-agent argumentation-based dialogue, and [8] for the definitions of various dialogue types). We treat the sender and the receiver as two independent information sources, which try to expand, update, and revise their beliefs through communication. Instead of adopting the computationally hard theory of trust [9], we consider three communicative relations between the agents involved: *communication with authority*, *peer to peer communication* and *communication with subordinate*.

To model phenomena such as lack and inconsistency of information, a commonly used logic is the four-valued logic proposed in [10]. However, as discussed, e.g., in [11, 12], the approach of [10] is problematic. In fact, in areas we focus on it often delivers results deviating from intuitions. Our approach is strongly influenced by ideas underlying the 4QL query language [13, 14] which does not share such problems. This natural four-valued model of interaction yields multiple new cognitive situations. Therefore we distinguish six interaction types and analyze them one by one, providing a sort of semantics of four selected speech acts: *assert*, *concede*, *request* and *challenge*. It is given in terms of triples consisting of *preconditions*, *speech acts* and *complex post actions*. Along with defining rules for perceiving speech acts, we indicate their detailed impact on the receiver's informational stance.

The paper is structured as follows. Section 2 is devoted to a four-valued logic used throughout the paper and provides basic information on 4QL. Section 3 discusses the technical contribution, illustrated by an example in Section 4. Section 5 concludes the paper.

2. 4QL: an Implementation Tool

In what follows all sets are finite except for sets of formulas.

We deal with the classical first-order language over a given vocabulary without function symbols and assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols. A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with $\bar{\tau} \in (\text{Const} \cup \text{Var})^k$, where k is the arity of R . *Ground literals over Const*, denoted by $\mathcal{G}(\text{Const})$, are literals without variables, with all constants in *Const*.

If $\ell = \neg R(\bar{\tau})$ then $\neg \ell \stackrel{\text{def}}{=} R(\bar{\tau})$. Let $v : \text{Var} \rightarrow \text{Const}$ be a *valuation of variables*. For a literal ℓ , by $\ell(v)$ we mean the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$. The semantics of propositional connectives is summarized in Table 1.

Table 1. Truth tables for \wedge , \vee , \rightarrow and \neg (see [12–14]).

\wedge	f	u	i	t	\vee	f	u	i	t	\rightarrow	f	u	i	t	\neg	f	t
f	f	f	f	f	f	f	u	i	t	f	t	t	t	t	f	t	
u	f	u	u	u	u	u	u	i	t	u	t	t	t	t	u	u	
i	f	u	i	i	i	i	i	i	t	i	f	f	t	f	i	i	
t	f	u	i	t	t	t	t	t	t	t	f	f	t	t	t	f	

Note that the definitions of \wedge and \vee reflect minimum and maximum w.r.t. the ordering:

$$f < u < i < t. \quad (1)$$

Definition 1 The *truth value* of a literal ℓ w.r.t. a set of ground literals L and valuation v , denoted by $\ell(L, v)$, is defined as follows:

$$\ell(L, v) \stackrel{\text{def}}{=} \begin{cases} \mathbf{t} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathbf{i} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathbf{u} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathbf{f} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$

△

For a formula $\alpha(x)$ with a free variable x and $c \in \text{Const}$, by $\alpha(x)_c^x$ we understand the formula obtained from α by substituting all free occurrences of x by c . Definition 1 is extended to all formulas in Table 2, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 1.

Table 2. Semantics of first-order formulas.

- if α is a literal then $\alpha(L, v)$ is defined in Definition 1;
- $(\neg\alpha)(L, v) \stackrel{\text{def}}{=} \neg(\alpha(L, v));$
- $(\alpha \circ \beta)(L, v) \stackrel{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\};$
- $(\forall x \alpha(x))(L, v) = \min_{a \in \text{Const}} (\alpha_a^x)(L, v)$, where min is the minimum w.r.t. ordering (1);
- $(\exists x \alpha(x))(L, v) = \max_{a \in \text{Const}} (\alpha_a^x)(L, v)$, where max is the maximum w.r.t. ordering (1).

From several languages designed for programming BDI agents (for a survey see, e.g., [15]), none directly addresses belief formation, in particular nonmonotonic or de-feasible reasoning techniques. 4QL enjoys tractable query computation and captures all tractable queries. It supports a modular and layered architecture, providing simple, yet powerful constructs for expressing nonmonotonic rules reflecting “lightweight” versions of many known formalisms [13].

Definition 2 By a *rule* we mean any expression of the form:

$$\ell :- b_{11}, \dots, b_{1i_1} \mid \dots \mid b_{m1}, \dots, b_{mi_m}. \quad (2)$$

where $\ell, b_{11}, \dots, b_{1i_1}, \dots, b_{m1}, \dots, b_{mi_m}$ are (negative or positive) literals and ‘,’ and ‘|’ abbreviate conjunction and disjunction, respectively. By a *fact* we mean a rule of the form ‘ $\ell :- .$ ’. Facts are abbreviated to ‘ $\ell.$ ’. A finite set of rules is called a 4QL *program*. △

Definition 3 Let a set of constants, Const , be given. A set of ground literals L with constants in Const is a *model of a set of rules S* iff for each rule (2) and any valuation v mapping variables into constants in Const , we have that:

$$(((b_{11} \wedge \dots \wedge b_{1i_1}) \vee \dots \vee (b_{m1} \wedge \dots \wedge b_{mi_m})) \rightarrow \ell)(L, v) = \mathbf{t},$$

where it is assumed that the empty antecedent takes the value \mathbf{t} in any interpretation. △

The semantics of 4QL is defined by well-supported models [13, 14]. Intuitively, a model is *well-supported* if all derived literals are supported by a reasoning grounded in facts. For any set of rules there is a unique such model and it can be computed in polynomial time [14].

3. Solution Details

3.1. Drawing upon Speech Acts Theory

Austin's observation that some utterances cannot be verified as true or false [16] led to the first division of speech acts into *constatives*, which can be assigned a logical truth value, and the remaining group of *performatives*. Searle created their most popular taxonomy, identifying: *assertives*, *directives*, *commissives*, *expressives* and *declaratives* [17]. Austin specified also their effects on the attitudes and actions of the hearer. Various speech acts, viewed as *typical actions*, can be represented in dynamic logic, by characterizing their pre- and post-conditions. We define them in terms of the changes in agents' beliefs and actions (see also [3–5]).

There have been many approaches to defining semantics of speech acts [1, 6], some based on Belnap's four-valued logic [18]. Still, some researchers view them as primitive notions [19]. Within the most popular *mentalistic* approach, reflected in languages such as KQML and FIPA ACL, speech acts are defined through their impact on agents' mental attitudes. The current paper clearly falls in that approach.

3.2. Epistemic Profiles and Communicative Relations

An essential question is how to realize heterogeneity of agents in multiagent systems. Clearly, being different, when seeing the same thing the agents may draw different conclusions. A notion of *epistemic profile* [20] explicitly models this problem. First, it defines the way an agent reasons (e.g., by the use of rules). Second, it permits expressing the granularity of reasoning (e.g., by varying the level of certain attributes or accuracy of rules expressing the modeled phenomena). Third, it also characterizes the manner of dealing with conflicting or lacking information by combining various forms of reasoning, including belief fusion, disambiguation of conflicting beliefs or completion of lacking information. The following definitions are adapted from [20], where also more intuition and examples can be found.

If S is a set then by $\text{FIN}(S)$ we understand the set of all finite subsets of S .

Definition 4 Let $\mathbb{C} \stackrel{\text{def}}{=} \text{FIN}(\mathcal{G}(\text{Const}))$ be the set of all finite sets of ground literals over the set of constants *Const*. Then:

- by a *constituent* we understand any set $C \in \mathbb{C}$;
- by an *epistemic profile* we understand any function $\mathcal{E} : \text{FIN}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over an epistemic profile* \mathcal{E} we mean $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$, where:
 - * $\mathcal{C} \subseteq \mathbb{C}$ is a nonempty set of constituents;
 - * $F \stackrel{\text{def}}{=} \mathcal{E}(\mathcal{C})$ is the *consequent* of $\mathcal{B}^{\mathcal{E}}$.

We alternate between the notions of the set of consequents and well-supported models.

Definition 5 Let \mathcal{E} be an epistemic profile. The *truth value of formula* α w.r.t. belief structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$ and valuation v , denoted by $\alpha(\mathcal{B}^{\mathcal{E}}, v)$, is defined by:

$$\alpha(\mathcal{B}^{\mathcal{E}}, v) \stackrel{\text{def}}{=} \alpha\left(\bigcup_{C \in \mathcal{C}} C, v\right).$$

△

Agents may react variously to the perceived information. We distinguish three types of *communicative relations*, considered from the receiver's perspective:

1. *communication with authority*: an agent (receiver) is prone to adopt the interlocutor's (sender, authority) theses. In strong disagreement, instead of abandoning its beliefs totally, it would rather investigate the reasons of the conflict, but in case of a mere discrepancy of opinions it would give up on its prior beliefs;
2. *peer to peer communication*: both parties are viewed as equally credible and important information sources, therefore nobody's opinion prevails a priori. This shows up when dealing with inconsistent information, which taints everything: whenever one party believes a proposition is inconsistent, the other party's prior beliefs do not matter. The peer, upon receiving information that introduces inconsistency to its beliefs, is obliged to comply with it. Therefore every discrepancy of opinions boils down to inconsistency,
3. *communication with subordinate*: when dealing with a less reliable source of information, the receiver with authority would not be willing to abandon its beliefs in favor of the interlocutor's. It would value its observations higher and protect true and false propositions from being infected by inconsistency. However, in case of strong disagreements, it engages in conflict resolution.

The communicative relation functions as a filter, determining when a new percept triggers **belief revision**. Importantly, it can be performed in polynomial time, due to the use of 4QL. Belief revision strategies, as expressed in 4QL, may vary from conservative to more drastic ones. This is however outside of the scope of this paper.

Introducing inconsistency as a first-class citizen entails a need of frequent **conflict resolution**. We distinguish two types of conflicts: *strong disagreements* and *mere discrepancies*. The first situation occurs when agents have contradictory opinions, the second, when one agent believes φ is inconsistent, and the other believes it is true or false.

When two agents first engage in communication they form a virtual group, for which we can define epistemic profiles as well. This is however outside of the scope of this paper (but see [20]). Information acquired via speech acts is then stored in the set of group level constituents, which are naturally transformed into consequents according to the epistemic profile. They represent the **beliefs shared** by the agents via communication.

3.3. A Four-valued Characterization of Speech Acts

We define speech acts by specifying their *preconditions* and *complex post-actions*. By:

$$\{precondition\} \langle speech\ act \rangle [complex\ post-action] \quad (3)$$

we mean that performing a *speech act* in the presence of the *precondition* triggers the *complex post-action*. We detail their semantics in the form of tables, in which, depending on the type of communicative relation and the preconditions, the shape of the speech act and the resulting complex post-action is determined. For simplicity, we number the respective communicative relation types, where: 1 signifies communication with an authority, 2 signifies peer to peer communication, and 3 communication with subordinate.

The precondition is in this case the receiver's valuation of the proposition in question. For clarity, throughout this section we employ the following notation:

$$v_R(\alpha) \stackrel{\text{def}}{=} \alpha(\mathcal{B}^{\mathcal{E}}, v), \text{ where } \mathcal{E} \text{ is agent } R \text{'s epistemic profile};$$

$$v'_R(\alpha) \stackrel{\text{def}}{=} \alpha(\mathcal{B}^{\mathcal{E}'}, v), \text{ where } \mathcal{E}' \text{ is agent } R \text{'s epistemic profile after performing belief revision on } \mathcal{E}.$$

3.3.1. Assertions

$\text{assert}_{S,R}(\alpha, x)$ stands for agent S (sender) telling agent R (receiver) that its valuation of α is x . Let us focus on different configurations, taking into account both communicative relations between the agents and their valuations of α :

1. *Perceiving inconsistent information*, where agents engaged in peer-to-peer communication or communication with authority relations, adopt the interlocutors' theses about inconsistent information. Authorities, however, adopt such a thesis only if it was already inconsistent or unknown. In both cases a *concede* is sent as an acknowledgement, but only in the latter case, belief revision takes place.
2. *Perceiving previously unknown information*, which leads to adopting it by the receiver regardless the communicative relation. Unless the new information is also unknown (see below), belief revision takes place and a *concede* is sent.
3. *Perceiving information that is unknown*, which is ignored regardless the communicative relation, because by default all information is unknown. Therefore, taking action under such circumstances would be redundant.
4. *Perceiving previously inconsistent information*, which depends on the communicative relation. In communication with authority, the sender's belief (unless unknown) overrides the receiver's. A belief revision takes place if necessary, that is, unless their beliefs are equal, and a *concede* is sent. In the two remaining cases the message is ignored, unless it is also inconsistent. If so, a *concede* is sent.
5. *Perceiving compatible information*, where "compatible" means that both agents have exactly the same valuation of the proposition. This yields no belief revision, but in all cases but one, the assertion is acknowledged by sending a *concede*. In the one case of unknown information such a perception is ignored.
6. *Perceiving contradictory information*, where, regardless communicative relation, whenever one agent believes a proposition is true and the other believes the contrary, they must come to an agreement by means of a *challenge* speech act. This may succeed, leading to adopting the sender's thesis, or fail, with no direct effect on the interlocutors. Notice, that upon receiving $\text{assert}_{S,R}(\alpha, x)$, the speech act $\text{challenge}_{R,S}(\alpha, x)$ stands for agent R asking agent S : "why does your valuation of α equal x ?". For more details on the semantics of challenges see Subsubsection 3.3.4 at the end of this subsection.

In Table 3, all complex post-actions of assertions discussed in cases 1-6 are summarized.

3.3.2. Requests

$\text{request}_{S,R}(\alpha)$ stands for agent S requesting agent R to provide information about α . After such a request, the sender must wait for a reply, while the receiver should reply with what he knows about α :

$$\{v_R(\alpha) = x\} \langle \text{request}_{S,R}(\alpha) \rangle [\text{assert}_{R,S}(\alpha, x)].$$

The sender, after receiving the response, behaves according to the rules for assertions.

3.3.3. Concessions

$\text{concede}_{S,R}(\alpha, x)$ stands for agent S 's communicating its agreement about the valuation of α . Importantly, only concessions about compatible valuations are considered, others

Table 3. Perceiving information.

Type	Precondition	Speech Act	Complex Post-Action ³
1, 2	$v_R(\alpha) = \mathbf{f}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$v'_R(\alpha) = \mathbf{i}; \text{concede}_{R,S}(\alpha, \mathbf{i})$
3	$v_R(\alpha) = \mathbf{f}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$v'_R(\alpha) = \mathbf{i}; \text{concede}_{R,S}(\alpha, \mathbf{i})$
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$\text{concede}_{R,S}(\alpha, \mathbf{i})$
1, 2	$v_R(\alpha) = \mathbf{t}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$v'_R(\alpha) = \mathbf{i}; \text{concede}_{R,S}(\alpha, \mathbf{i})$
3	$v_R(\alpha) = \mathbf{t}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{f})$	$v'_R(\alpha) = \mathbf{f}; \text{concede}_{R,S}(\alpha, \mathbf{f})$
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$v'_R(\alpha) = \mathbf{i}; \text{concede}_{R,S}(\alpha, \mathbf{i})$
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{t})$	$v'_R(\alpha) = \mathbf{t}; \text{concede}_{R,S}(\alpha, \mathbf{t})$
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{f})$	$v'_R(\alpha) = \mathbf{f}; \text{concede}_{R,S}(\alpha, \mathbf{f})$
2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{f})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$\text{concede}_{R,S}(\alpha, \mathbf{i})$
1	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{t})$	$v'_R(\alpha) = \mathbf{t}; \text{concede}_{R,S}(\alpha, \mathbf{t})$
2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{t})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{assert}_{S,R}(\alpha, \mathbf{f})$	$\text{concede}_{R,S}(\alpha, \mathbf{f})$
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{assert}_{S,R}(\alpha, \mathbf{u})$	-
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{assert}_{S,R}(\alpha, \mathbf{i})$	$\text{concede}_{R,S}(\alpha, \mathbf{i})$
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{assert}_{S,R}(\alpha, \mathbf{t})$	$\text{concede}_{R,S}(\alpha, \mathbf{t})$
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{assert}_{S,R}(\alpha, \mathbf{f})$	$\text{challenge}_{R,S}(\alpha, \mathbf{f})$
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{assert}_{S,R}(\alpha, \mathbf{t})$	$\text{challenge}_{R,S}(\alpha, \mathbf{t})$

are ignored, as indicated in Table 4. Concession is more of an acknowledgement, as no belief revision on the individual level takes place here. Instead, α with a valuation x is added to the agents' virtual group's set of constituents (see Subsection 3.2) and (possibly) a belief revision on the virtual group's level occurs.

3.3.4. Challenges

$\text{challenge}_{S,R}(\alpha, x)$ stands for S 's communicating its contradictory stance with respect to R 's opinions regarding α (x is either \mathbf{t} or \mathbf{f} here). Inspired by [8], a challenge is in fact a request to provide a proof of the receiver's stance towards α together with an implicit assertion of the contradictory stance towards α . To restrict emitting redundant information we make this assertion implicit.

$$\begin{aligned}\text{challenge}_{S,R}(\alpha, \mathbf{t}) &\equiv \text{request}_{S,R}(\text{assert}_{R,S}(\text{PROOF}(\alpha, \mathbf{t}))) \\ \text{challenge}_{S,R}(\alpha, \mathbf{f}) &\equiv \text{request}_{S,R}(\text{assert}_{R,S}(\text{PROOF}(\alpha, \mathbf{f})))\end{aligned}$$

The proof in question depends on the structure of α and might represent just the last rule used to derive α (or a choice of rules if there were several ways to achieve α). If, for an atomic α , its negation $\neg\alpha$ is a fact, there is no way to prove that α is true, and the challenged agent who received $\text{challenge}_{S,R}(\alpha, \mathbf{t})$ should reply with a special

Table 4. Requests, concessions and challenges. Cases not included in the table are ignored.

Type	Precondition	Speech Act	Complex Post-Action
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{request}_{S,R}(\alpha)$	$\text{assert}_{R,S}(\alpha, \mathbf{f})$
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{request}_{S,R}(\alpha)$	$\text{assert}_{R,S}(\alpha, \mathbf{u})$
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{request}_{S,R}(\alpha)$	$\text{assert}_{R,S}(\alpha, \mathbf{i})$
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{request}_{S,R}(\alpha)$	$\text{assert}_{R,S}(\alpha, \mathbf{t})$
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{concede}_{S,R}(\alpha, \mathbf{f})$	add α to group ^a constituents ^b
1, 2, 3	$v_R(\alpha) = \mathbf{u}$	$\text{concede}_{S,R}(\alpha, \mathbf{u})$	add α to group constituents
1, 2, 3	$v_R(\alpha) = \mathbf{i}$	$\text{concede}_{S,R}(\alpha, \mathbf{i})$	add α to group constituents
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{concede}_{S,R}(\alpha, \mathbf{t})$	add α to group constituents
1, 2, 3	$v_R(\alpha) = \mathbf{f}$	$\text{challenge}_{S,R}(\alpha, \mathbf{t})$	$\text{assert}_{R,S}(\text{PROOF}(\alpha, \mathbf{t}))$
1, 2, 3	$v_R(\alpha) = \mathbf{t}$	$\text{challenge}_{S,R}(\alpha, \mathbf{f})$	$\text{assert}_{R,S}(\text{PROOF}(\alpha, \mathbf{f}))$

^aThe virtual group consisting of R and S .

^bAdding facts to the set of constituents may trigger belief revision on the virtual group level.

symbol $\text{assert}_{R,S}(\alpha, \perp)$ for “I give up” (cf. [8]). This counts as agent’s R failure to prove α . An agent who receives a challenge should react according to the rules for requests. Challenges per se do not yield belief revision, the true impact on agents’ beliefs is achieved by the assertions initiated by them. However, when a (possibly deeply nested) challenge folds, if it was successful, an acknowledgement must be sent. Therefore whenever an agent sends $\text{challenge}_{S,R}(\alpha, x)$, if at some point α becomes x it triggers sending a concede by S . This is a sign of a challenge ending successfully for R . Otherwise a challenge fails. Notice that even a failed challenge might have caused some belief revision if some of the assertions have been acknowledged.

4. An Example

To demonstrate perceiving speech acts in the four-valued framework we use the 4QL interpreter for modeling and for computing well-supported models.⁴ For simplicity, we focus on the case when the sender S is an authority and the receiver R a subordinate.

A rescue-agent (receiver) is capable of putting out fire and detoxicating an area. Suppose at the moment he is aware of the fire but due to faulty sensors cannot sense the heat. A simplified program in 4QL referring to this situation is presented in Figure 1. Then, the following unique well-supported model B_{R_1} represents agent R ’s belief base:

$$B_{R_1} = \{\text{heat}, \neg\text{heat}, \text{detox}, \neg\text{detox}, \text{fire}, \text{busy}, \text{extinguish}\}. \quad (4)$$

On the basis of B_{R_1} we know that agent R is inconsistent about *heat* and *detox*. Literals *fire*, *busy* and *extinguish* are true. Literals absent in the model are unknown.

It starts when agent S , an authority, asserts to R that *heat* is true (see (a) in Figure 2). In R ’s belief base B_{R_1} , *heat* is inconsistent. Therefore, after perceiving such an assertion, according to the rules (see Table 3), agent R adopts S ’s belief. In the course of belief revision, the program shown in Figure 1 is changed so that $\neg\text{heat}$ is no longer a fact. Instead, we put *heat* as a new fact and compute anew well-supported model B_{R_2} :

$$B_{R_2} = \{\text{heat}, \neg\text{detox}, \text{fire}, \text{busy}, \text{extinguish}, \text{risky}\}. \quad (5)$$

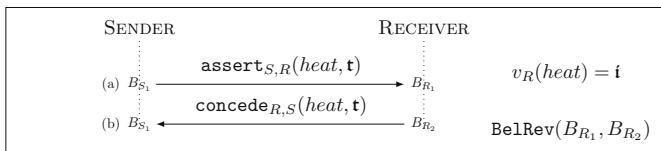
Finally, agent R answers with a concession (see (b) in Figure 2), acknowledging that it shares a belief about *heat* being true. Notice, that if R ’s valuation of *heat* would have

⁴The interpreter, developed by P. Spanily, can be downloaded from <http://www.4ql.org/>.

```

module r:
    relations: a(literal).
    rules:
        a(risky) :- a(poison) | a(heat).
        -a(risky) :- -a(poison), -a(heat).
        a(heat) :- a(fire).
        -a(poison) :- a(safe).
        a(busy) :- a(detox) | a(extinguish).
        -a(detox) :- a(heat).
        -a(heat) :- a(lowtemp).
        -a(fire) :- a(lowtemp).
    facts:
        a(fire).
        -a(heat).
        a(extinguish).
end.

```

Figure 1. Example of a 4QL program.**Figure 2.** Perceiving compatible information.

been false in the first place (in B_{R_1}), he would have to challenge the sender's assertion. Then, S would have to provide a proof.

5. Conclusions

Agents need to act in uncertain and dynamic environments, where they receive information from multiple sources. A new four-valued paraconsistent logic [13, 14] appears to be cut out for this situation. Nonmonotonic logic, in turn, allows for drawing conclusions that typically hold, but not necessarily always. Their combination, implemented in 4QL, has already been shown to model agents' individual and group beliefs. Agents' reasoning schemas are formalized in terms of rules in the chosen four-valued knowledge-based framework, belief structures and epistemic profiles [20]. A great bonus of 4QL is that queries can be computed in polynomial time. This tractability stands in stark contrast to the usual two-valued approaches to group interactions, where EXPTIME completeness of satisfiability problems is a common hindrance [21].

This paper is a first step in a research program that combines dialogue theory and argumentation theory with the new four-valued approach to modeling multi-agent interactions. We consider three types of communication: from an authority, from a peer, and from a subordinate. In each type of communication, the speech acts are considered from the mentalistic perspective, as expressed in the triple: precondition, speech act, complex post-action. Along with defining rules for perceiving speech acts, we indicated how communication influences agents' beliefs.

Our example of a simple dialogue between a rescue-agent and its boss shows how speech acts and agents' reasoning rules naturally combine in the framework of 4QL, leading to intuitive conclusions while maintaining tractability. Thus, a foundation has

been laid for extending the four-valued approach to modeling more complex dialogues and argumentations between agents reasoning in uncertain and dynamic environments. This will be the subject of the forthcoming paper.

References

- [1] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152:157–206, 2005.
- [2] K. Budzynska, M. Kacprzak, and P. Rembelski. Perseus. software for analyzing persuasion process. *Fundamenta Informaticae*, 93(1-3):65–79, 2009.
- [3] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9:145–158, 2001.
- [4] B. Dunin-Kęplicz, A. Strachocka, and R. Verbrugge. Deliberation dialogues during multi-agent planning. *ISMIS*, volume 6804 of *LNCS*, pp. 170–181. Springer, 2011.
- [5] B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley, Chichester, 2010.
- [6] Simon Parsons and Peter McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12:415–439, 2003.
- [7] C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agent Systems*, 11(2):173–188, 2005.
- [8] D.N. Walton and E.C.W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.
- [9] C. Castelfranchi. The social nature of information and the role of trust. *International Journal of Cooperative Information Systems*, 11(3):381–403, 2002.
- [10] N.D. Belnap. A useful four-valued logic. In G. Epstein and J.M. Dunn, editors, *Modern Uses of Many Valued Logic*, pp. 8–37. Reidel, 1977.
- [11] D. Dubois. On ignorance and contradiction considered as truth-values. *Logic Journal of the IGPL*, 16(2):195–216, 2008.
- [12] A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.
- [13] J. Małuszyński and A. Szałas. Living with inconsistency and taming nonmonotonicity. In O. de Moor, G. Gottlob, T. Furche, and A. Sellers, editors, *Datalog Reloaded*, volume 6702 of *LNCS*, pp. 384–398. Springer-Verlag, 2011.
- [14] J. Małuszyński and A. Szałas. Partiality and inconsistency in agents’ belief bases. In *Proc. AMSTA’13: Agents and Multi-agent Systems Technologies and Applications*. IOS Press, 2013.
- [15] V. Mascardi, D. Demergasso, and D. Ancona. Languages for programming BDI-style agents: an overview. In F. Corradini, F. De Paoli, E. Merelli, and A. Omicini, editors, *WOA 2005 - Workshop From Objects to Agents*, pp. 9–15, 2005.
- [16] J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, second edition, 1975. Edited by J. O. Urmsen and M. Sbisa.
- [17] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, 1969.
- [18] H. Lebbink, C. Witteman, and J.-J. Meyer. Dialogue games for inconsistent and biased information. *Electronic Notes in Theoretical Computer Science*, 85(2):134–151, 2004.
- [19] H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.
- [20] B. Dunin-Kęplicz and A. Szałas. Epistemic profiles and belief structures. In G. Jezic, M. Kusek, N.T. Nguyen, R.J. Howlett, and L.C. Jain, editors, *KES-AMSTA*, volume 7327 of *LNCS*, pp. 360–369. Springer-Verlag, 2012.
- [21] M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz. Complexity issues in multiagent logics. *Fundamenta Informaticae*, 75(1-4):239–262, 2007.

Social and Business Issues

The Agent-Based Diffusion Model on a Combined Social Network

Setsuya KURAHASHI¹

Graduate School of Business Sciences, University of Tsukuba, Japan

Abstract. In this paper, we discuss how neighbours affect consumer behaviour over diffusion of innovation. We propose an agent-based model of diffusion on an online social network, which has both “scale-free” and “regular” properties. The findings of the study are the following points: 1) the informative effect can cause a take-off, but it is not sufficient to reach the completion of diffusion, 2) the combination of the informative and normative effects can easily bring a take-off, 3) the informative effect makes information propagate fast, and so does the normative effect over a network that has characteristics of scale-free and high cluster, 4) the selective approach to low-degree consumers are more efficient to see the complete diffusion.

Keywords. social simulation, word-of-mouth, social networks, marketing, selective advertisement

1. Introduction

Companies have been able to take a wide variety of advertisement strategies in addition to mass-media advertising. It has previously been done, such as by launching campaign websites, and viral marketing on the Internet services including social network services(SNS). Trends (booms) that break out of unexpected movements or places are considered that the word-of-mouth effects have a significant influence when compared to running ordinary advertisements. Word-of-mouth effects are interactions between consumers. Utilisation of such word-of-mouth effects has been examined today. Evidently each advertisement through the media has had a certain level of effect; however, the reaction mechanism of consumers on the Internet has not been clarified. The word-of-mouth effects have had a significant influence on the Internet [1]. This has brought difficulties when it comes to making decisions on websites.

Through this research, we show that there are limitations when treating consumer interactions such as word-of-mouth communication in an integrated fashion in diffusion. By doing so, we try to examine the mechanism by which the opinions of surrounding consumers affect one's own decision-making process.

This paper is organized as follows. In Section 2, we describe previous researches. Section 3 proposes Agent-based Diffusion Model, and Section 4 describes simulation results of diffusion processes and selective advertising effectiveness. In section 5, the informative effect and the normative effect are discussed. Section 6 proposes ideas for future research work and concludes the paper.

¹Corresponding Author: Setsuya Kurahashi, University of Tsukuba, Graduate School of Business Sciences, 3-29-1 Otsuka, Bunkyo, Tokyo, Japan; E-mail: kurahashi.setsuya.gf@u.tsukuba.ac.jp.

2. Previous Research

The representative research about diffusion is a study using mathematical models, including the Bass model. The solution to the differential equation of the Bass model forms the sigmoid curve of diffusion [2], and its take-off is caused by the imitation effect. Take-off is defined as critical mass which is described as the existence of a sufficient amount of adopters of an innovation or a product in a diffusion process. Once take-off occurs in a diffusion process, the continued adoption keeps self-sustaining growth by the interactive actions among the members of the social system.

Recently, based on agent-based models of diffusion, studies have been conducted in which networks, where consumer interaction is generated, were explicitly provided. Goldenberg et al. [3,4,5] discussed the role of a hub in a scale-free network by using ABS. Delre et al. made a study of diffusion in a small-world network² by utilising ABS and showed that the small-world feature of the network and consumer heterogeneity accelerate diffusion [6]. They adopted the threshold model [7,8] in which the probability of change in consumer behaviour (status transition) increases in a discontinuous manner according to the percentage of neighbouring adopters, in the consumer's decision-making process of accepting a product.

Similarly, Watts et al. adopted the threshold model in which the probability of change in the consumer behaviour generates according to the percentage of neighbouring adopters [9,10]. Either of these models proposes a model in harmony with individual networks to be used, and it is impossible to explain the phenomenon of diffusion where each network is replaced. Delre et al. actually indicated that their own model could not be applied to a scale-free network³ [11]. On the other hand, some studies of viral marketing based in social networking have been issued recent years [12,13]. These studies describe more realistic phenomenon but are not concerned about normative and informative effects on a combined network.

3. Agent-based Diffusion Model

As a decision-making model for consumers, this model defines the probability of transitioning from the susceptible status to the infected status. In this model, the probability of status transition of a consumer i is P_i . Consumers with a high probability of status transition easily change their behaviour.

In this research, it is supposed that the informative effect and the normative effect have an influence on the change in consumer behaviour. The informative effect indicates the possibility of access to information by means of searching, which is considered to be the influence exerted by the number of neighbouring adopters (*number-of-exposure rule*). On the other hand, the normative effect indicates the possibility of communication in a highly-clustered relationship, which is the influence provided by the percentage of the neighbouring adopters (*proportion-of-households-threshold rule*). Based on this concept, the probability of change in behaviour P_i is defined below.

²A small-world network has a small average shortest path length, but also a clustering coefficient significantly higher than expected by random chance.

³A scale-free network is a network whose degree distribution follows a power law.

$$P_i = \alpha \cdot inf_i + (1 - \alpha) \cdot norm_i \quad (1)$$

$$inf_i = \begin{cases} 1 & (N_{i,adopted} \geq \xi_i) \\ 0 & (otherwise) \end{cases} \quad (2)$$

$$norm_i = \begin{cases} 1 & \left(\frac{N_{i,adopted}}{N_{i,neighbour}} \geq \phi_i \right) \\ 0 & (otherwise) \end{cases} \quad (3)$$

In this equation, inf_i and $norm_i$ are the variables that indicate the informative effect and the normative effect, respectively. $N_{i,neighbour}$ and $N_{i,adopted}$ indicate the number of neighbouring consumers of the consumer i and the number of the neighbouring adopters, respectively. ξ_i is the threshold of the informative effect, while a change in behaviour is more likely to occur where the number of neighbouring adopters exceeds this threshold. Similarly, ϕ_i is the threshold of the normative effect, and change in behaviour also is more likely to occur when the percentage of the neighbouring adopters exceeds this threshold. α is the weight to the informative effect and the normative effect. The utility of using a product consists of an individual part (informative effect) and a social effect part (normative effect) [14]. The individual part expresses the difference between personal preferences of a consumer for each product and the product dimension.

3.1. Network of Interactions among Consumers

The previous studies have confirmed that human-relationship networks on the Internet are scale-free and highly clustered networks [15,16]; however, it is impossible to generate such networks by using the existing mathematical models. For this reason, the network desired in this research was formed by synthesising each network generated by using the Watts-Strogatz (WS) model and the Barabasi-Albert (BA) model. First, the regular network with the degree of 4 (Table 1: Regular) was generated and the scale-free network (Table 1: ScaleFree) was generated by using the BA model. With these networks, the network ScaleFreeC (SC) was created by obtaining the logical sum of the corresponding link of each network and then overlapping these two networks. The process of diffusion on the created network, ScaleFreeC, was then observed. Table 1 shows the characteristics of each network (the average degree, the average path length, and the clustering coefficient). Figure 1 shows an example of a combined social network SC (The number of nodes : 20).

Table 1. Networks (SC: ScaleFreeC, RG:Regular, SF:ScaleFree)

Network	Method	Average degree	Average path length(L)	Clustering coeff.(C)
SC	SF + RG	5.998	4.24	0.216
RG	WS[2k=4, p=0]	4.000	125.38	0.400
SF	BA	1.998	7.50	0.000

3.2. Selective Advertising Effectiveness

How do you spread the information more widely? According to the two-step flow theory in marketing, first information should reach opinion leaders with advertisements. Sec-

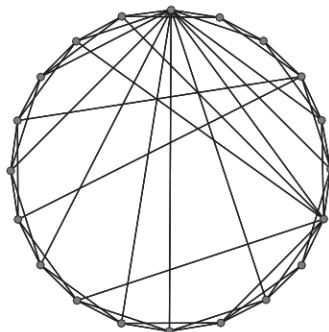


Figure 1. An example of a combined social network SC (The number of nodes : 20).

ondly, it is conveyed to other consumers who are followers. The simulation, however, shows that it is not sufficient. In the former section, it has been confirmed that the diffusion process progresses from take-off to completion only if both of the effects exist. Besides it has been also found that low-degree consumers should change high-degree consumers. Therefore another experiment is carried out based on selective advertising effectiveness.

In this experiment, we adopt power weight on marketing effort, which is selective advertising to consumers. To keep marketing effort constant, the probability of marketing effort P_{ei} is defined below. $P_{ei} = \varepsilon * \frac{degree_i^w}{\sum_i degree_i^w/n}$, where ε , $degree_i$, i , w , and n indicate advertising effectiveness, a degree of an agent, an agent, weight, and the number of agents respectively. Advertising effectiveness ε is set to a small value : 0.0001 to observe effects of not advertising but interaction in intermediate steps, and n is set to 1000. The following parameters impose the weight to low or high-degree consumers for the selective advertisement. The weight of selective advertisement approach to high-degree consumers : $w = 2.0$, the weight of selective advertisement approach to low-degree consumers : $w = -2.0$, the weight of non-selective advertisement approach to all consumers: $w = 0.0$. The selection effects on consumers are decided with these weights. To equalise the total amount of the probability P_{ei} of selective marketing effort in all agents, P_{ei} is normalised with the total number of degrees in all agents for each weight of selective advertisement approach.

4. Simulation

In this section, some simulation trial results are shown by changing the thresholds ξ_i, ϕ_i of the informative effect, which is the internal status of the consumer agent (the number of agents was set to 1000) in the proposed model, and the normative effect, and the networking of consumer-agent interactions. The iteration time that indicates the elapsed time was set to 1000. The previous transition of the number of adopters was mainly observed. Parameter α , which indicates the weight of the informative effect and the normative effect, was varied within the range from 0 to 1. The power weight w on marketing effort was set to 0. Given this, we confirmed the feature of information propagation from both effects in each network.

The proposed model only defines the interactions among consumers, and only with this definition the model does not form the sigmoid curve that indicates the diffusion process. For this reason, consumers are randomly selected based on a certain percentage and this causes change in behaviour regardless of their interactions. This is referred to as the seed (e), which is considered to be the effect of advertisements run by the mass media. Change in behaviour of consumers by means of the seed is caused time to time during the simulation performed.

Assuming the situation where the quantity of advertisements run is extremely small, we assigned a small value to the seed. This could cause some cases where diffusion does not reach completion. The following three points of the simulation results should be focused on: (1) whether the take-off due to consumer interactions would occur or not, (2) whether diffusion would reach completion or not, and (3) how the velocity for diffusion to reach completion could be accelerated. In many cases of services and products, the diffusion curves show the similar shapes, such as a personal computer, a car, a facsimile, and a mobile phone [17].

The diffusion curves rise rapidly when the local diffusion rate achieves over 10 to 20 percent, then the curves level off when achieving 60 to 70 percent of the rate. Regarding the number of neighbours influencing personal preferences of a consumer (informative effect), it is reported that 36 percent of a consumer's acquaintances send information about a service to the consumer as an opinion leader, a market maven or a leading consumer, where the number of acquaintances are distributed mostly between 1 to 10 [18]. Therefore, agent attributes, ξ and ϕ , were assigned so that they would form a uniform distribution in the range of [1, 4] and [0.2, 0.6], respectively.

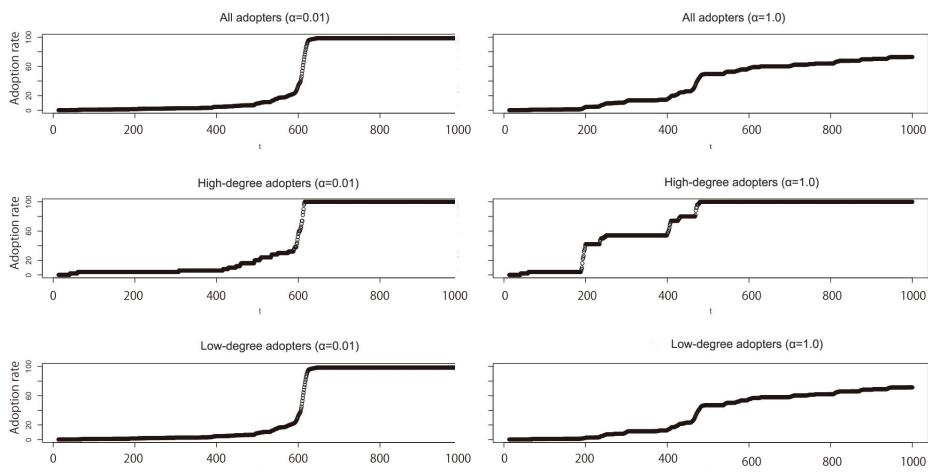


Figure 2. Diffusion curve of adopters. Left: $\alpha = 0.01$ Right: $\alpha = 1.00$

We examined the differences in diffusion where the parameter α was varied based on each of the probability functions of status transitions. In the case of only the informative effect, although a slight take-off occurs, diffusion does not reach completion by the specified time ($\alpha = 1.00$). In the case of only the normative effect, take-off does not occur.

cur ($\alpha = 0.00$). In contrast, where both of the effects are mixed, diffusion rapidly reaches completion from the take-off ($\alpha = 0.01$ to 0.09). Therefore α was set to 0.01.

4.1. Diffusion Process

In this section it is clarified that what nodes information reaches and how it affects the diffusion process with the informative effect and the normative effect. We confirm how much information reaches high-degree consumers (HD) and low-degree consumers (LD), where HD is defined as consumers who have 10 degrees or more and LD have less than 10 degrees. The number of degrees (=10) is just set for observation, therefore it is not a parameter for the model. It is supposed that HD have many connections to other consumers and can affect their decision making, while LD have less connections. It will show what kinds of people receive information with the informative effect and the normative effect.

Figure 2 shows the temporal transition of the rate of all the adopters (diffusion rate), the adopters of HD (degree ≥ 10), and the adopters of LD (degree < 10), respectively. Big differences are not seen until the iteration time gets to 200 because the conditions including the random seed except for the parameter α are the same. After some adopters appear along with the advertisement effect e , each ratio begins to change with α . In the case of only the informative effect ($\alpha = 1.00$, Figure 2:right), the adoption rate of HD is more than 40% around the time 200, while the adoption rate of LD does not follow it. It indicates that the informative effect is powerful for HD but not for LD because the effect is exerted by the number of neighbouring adopters.

In contrast, where both of the effects are mixed ($\alpha = 0.01$), diffusion rapidly proceeds after the take-off around the time 600, and then it reaches completion eventually in all of three figures (Figure 2:left). In the case of $\alpha = 0.01$, HD and LD change to the adoption condition simultaneously around the time from 400 to 600, it increases rapidly to reach completion around the time 600. From these figures, we arrive at the following conclusion. The informative effect can prompt to distribute information to HD, but it cannot have much of an effect on LD. The normative effect supports to the conveying of information to LD with the network having a high clustering coefficient, but it cannot have the same effect on HD. It means paths to transport information are different between both the effects. Although the diffusion process progresses from take-off to completion only if both the effects exist, it is not enough that high-degree consumers come to the adoption condition, it is also necessary for completion to get low-degree consumers to change their behaviour.

In Figure 3, the selective advertisement to mainly low-degree consumers (red line at the top of the graph) is more effective than to mainly high-degree consumers (green line at the top of the graph). Although the selective advertisement to mainly high-degree consumers shows the fastest take-off (green line in the middle of the graph), the most effective approach is the non-selective advertisement for all consumers (black line at the top of the graph). According to the experiment, the selective advertisement approach to high-degree consumers is only effective in the early period of the diffusion process, but the non-selective approach and the selective approach to low-degree consumers are more efficient to see the complete diffusion after the early stage.

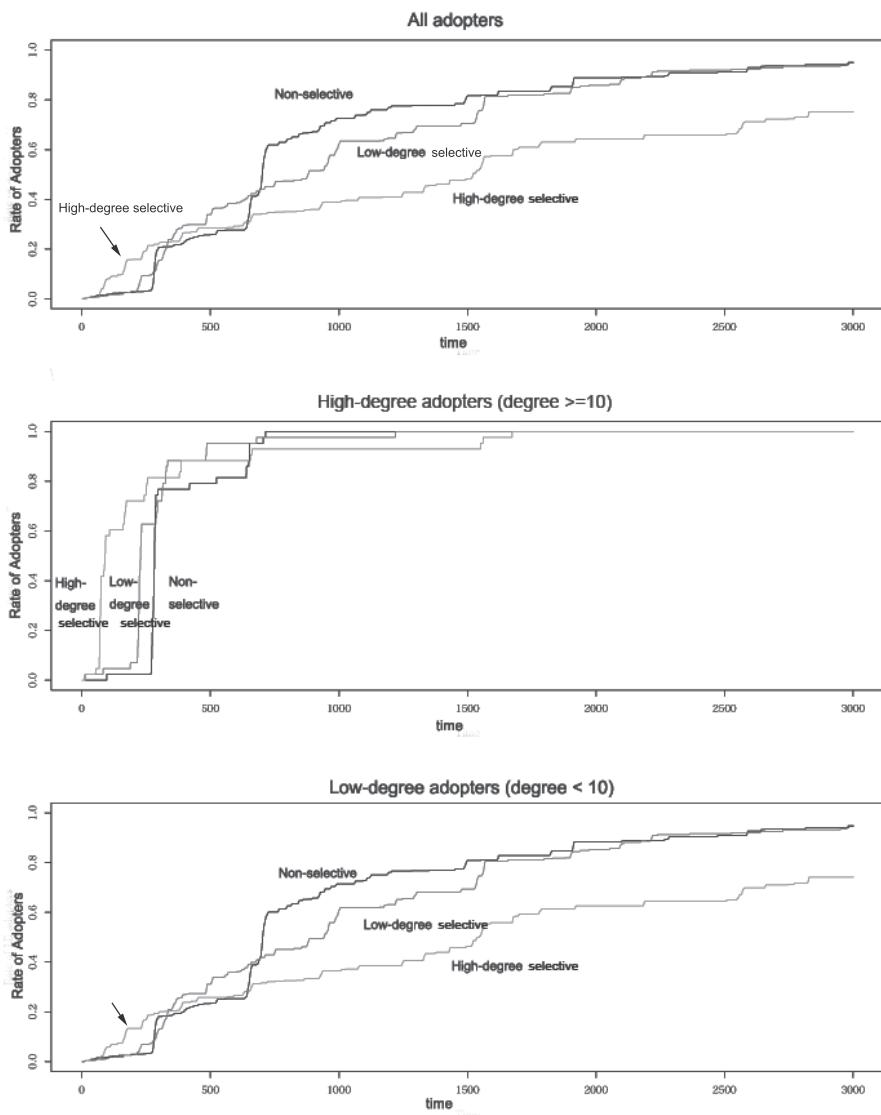


Figure 3. Diffusion curves of selective advertising effectiveness. The upper graph shows the rate of adopters in all consumers, the middle graph shows the rate of adopters in high-degree consumers, the lower graph shows the rate of adopters in low-degree consumers. Each line indicates the rate of adopters by the marketing effect of non-selective advertisement to all consumers (black line), advertisement to high-degree consumers (green line), advertisement to low-degree consumers (red line), respectively.

5. Discussion

In the previous section the proposed model was implemented and the conditions of the diffusion (information propagation) process were also clarified, especially under the situation where only a small quantity of advertisements exists, based on the simulation performed. With the simulation results discussed, this chapter examines the simulation

execution conditions and the results in the light of the simulation and the findings of the studies of consumer activity in order to show the following points. Reviewing the characteristics of rumour propagation based on the research about rumours, we compare our results with the interactions among consumers and the simulation results in the research field of marketing [1].

The existing research on diffusion adopted either of the following two models to be treated as the word-of-mouth model: the informative effect, in which the number of the neighbouring adopters affects change in consumer behaviour, and the normative effect, in which the percentage of the neighbouring adopters affect change in consumer behaviour. However, the characteristics of information propagation of each model are different, thus they are identifiable. The type of communication (interactions) among consumers includes the information-seeking type and the self-contained type. The information propagation paths provided by these communication types correspond to the propagation path of the informative effect and that of the normative effect, respectively.

It is confirmed that the selective advertisement approach to high-degree consumers is only effective in the early period for spreading word-of-mouth on the network, but non-selective approach and the selective approach to low-degree consumers are more efficient to see the complete diffusion after the early stage. Two-step flow theory has been used as one of marketing approaches for diffusion so far. It hypothesises that ideas flow from mass media to opinion leaders and from them to a wider population [19]. The results, however, indicate that marketers of companies should adopt a two-ways strategy which switches approaches between high-degree consumers in the early stage and all consumers after the stage.

6. Conclusion

In this paper we proposed the agent-based diffusion model, and the simulation we performed indicated that there are different paths by which the informative effect and the normative effect convey information. The informative effect indicates the exploratory action of gaining information, whereas the normative effect indicates the imitation effect that works on how consumers feel and try to keep up with other consumers, network externalities, and explanations from society. The traits and the paths of information propagation actually differ. Therefore, there is a limitation when treating either of these two effects as the word-of-mouth effect, just as done by the previous studies on word-of-mouth communication by using simulation.

This paper also showed that it is inadequate to think that opinion leaders, connected with numerous other consumers, only adopt a product and transmit the information of usability impressions to other consumers in order to trigger diffusion on online human-relationship networks. Rather, diffusion is promoted entirely by active communication among non-opinion leaders which have received such information from opinion leaders.

References

- [1] T. Yoshida, M. Hasegawa, T. Gotoh, H. Iguchi, K. Sugioka, K. Ikeda: Consumer Behavior Modeling Based on Social Psychology and Complex Network, *CEC-EEE2007*, 2007.

- [2] F. M. Bass: A New Product Growth for Model Consumer Durables, *Management Science*, Vol. 15, No. 5, pp. 215–227, 1969.
- [3] J. Goldenberg, B. Libai, E. Muller: Using complex systems analysis to advance marketing theory development: Modelling heterogeneity effects on new product growth through stochastic cellular automata, *Academy of Marketing Science Review*, Vol. 2001, No. 9, 2001.
- [4] J. Goldenberg, B. Libai, E. Muller: Riding the saddle, how cross-market communications creates a major slump in sales, *Journal of Marketing*, Vol. 66, pp. 1–16, 2002.
- [5] J. Goldenberg, S. Han, D. R. Lehmann, J. W. Hang: The role of hubs in the adoption processes, *Journal of Marketing*, Vol. 73, pp. 1–13, 2009.
- [6] S. A. Delre, W. Jager, M. A. Janssen: Diffusion Dynamics in Small-World Networks with Heterogeneous Consumers, *Computational and Mathematical Organization Theory*, Vol. 13, pp. 185–202, 2007.
- [7] M. Granovetter: The strength of weak ties, *American Journal of Sociology*, Vol. 78, No. 6, pp. 1360–1380, 1973.
- [8] M. Granovetter: Threshold Models of Collective Behavior, *American Journal of Sociology*, Vol. 83, No. 6, pp. 1420–1443, 1978.
- [9] S. Milgram: The Small World Problem, *Psychology Today*, Vol. 2, pp. 60–67, 1967.
- [10] D. Watts, P. Dodds: Influentials, Networks, and Public Opinion Formation, *Journal of Consumer Research*, Vo. 34, No. 4, pp. 441–458, 2007.
- [11] S. A. Delre, W. Jager, T. H. A. Bijmolt, M. A. Janssen: Targeting and timing promotional activities: An agent-based model for the takeoff of new products, *Journal of Business Research*, Vol. 60, No. 8, pp. 826–835, 2007.
- [12] V. Podobnik, G. Vedran, T. Krunoslav, J. Gordan: Group-Oriented Service Provisioning in Next Generation Network, *Innovations in Multi-Agent Systems and Applications / Srinivasan, Dipti; Jain, Lakhmi C., editor(s)*. Berlin Heidelberg: Springer-Verlag, pp. 277–298, 2010
- [13] L. Vrdoljak, V. Podobnik, J. Gordan: Forecasting Consumer Interest in New Services using Semantic-aware Prediction Model: the Case of YouTube Clip Popularity, *Lecture Notes in Computer Science*. 7327, pp. 454–463, 2012
- [14] M. A. Janssen, W. Jager: Simulating Market Dynamics: Interactions between Consumer Psychology and Social Networks, *Artificial Life*, Vol. 9, pp. 343–356, 2003.
- [15] Y. Matsuo, Y. Yasuda: How relations are built within a SNS World - Social network analysis on Mixi, *Transactions of the Japanese Society for Artificial Intelligence*, Vol. 22, No. 5, pp. 531–541, 2007.
- [16] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, P. Bhattacharjee: Measurement and analysis of online social networks, *In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC'07*, pp. 29–42, 2007.
- [17] Cabinet Office, Monthly Consumer Confidence Survey covering all of Japan, Government of Japan, <http://www.esri.cao.go.jp/jp/stat/shouhi/shouhi.html>
- [18] K. Ikeda, editor: The Diffusion of Innovations through Word-of-Mouth and Social Networks: A Social Psychological Study combining Snowball Surveys and Multi-Agent Simulation, University of Tokyo Press, 2010
- [19] E. Katz: The Two-Step Flow of Communication: An Up-To-Date Report on a Hypothesis, *The Public Opinion Quarterly*, Vol. 21, No. 1, pp. 61–78, 1957.

Towards the Validation of Agent-based BPM Simulation

Roman ŠPERKA^{a,1}, Dominik VYMĚTAL^a and Marek SPIŠÁK^a

^a*Silesian University in Opava, School of Business Administration in Karviná,
 Department of Informatics, Karviná, Czech Republic*

Abstract. One of the efficient methods of business process simulation is the use of a multi-agent system. As on other cases, such simulation requires sufficient input data. However, in the case of business systems, real business data is not always available, therefore multi-agent systems often operate with randomly (or pseudo randomly) generated parameters. This method can also allow for unpredictable phenomena both inside the simulated system and in the environment. The motivation of this paper is to introduce novel business process simulation method based on multi-agent system and to validate the simulation results with certain randomly generated agent parameters. Normal distribution was used to model two agent parameters in the simulation. These parameters are the amount of goods to buy and the ability of the seller agents. Certain number of negotiating agents representing various important roles within the company business system is described. The obtained results demonstrate that using random number generation function from normal distribution can lead to proper simulation results in comparison with real data. This partial research is a last step to validate the innovative implementation of a decision support tool.

Keywords. BPM, random, control loop, distribution, JADE, multi-agent, simulation, modeling, business process

1. Introduction

Due to the ongoing impact of globalization, the importance of business systems modeling has recently undergone a period of rapid growth. The managements of business companies have to increase flexibility and tempo of decision-making in order to maintain pace with market developments. The complexity of business operations often does not allow for taking measures without knowing the impacts of such decisions. This is precisely where modeling and simulations demonstrate their importance (see e.g. [1]). While analytical modeling approaches are based mostly on mathematical theories, [2] our approach concentrates on experimental simulations. The generic business company used for these simulations, using the control loop paradigm [3-8], is presented in Figure 1. - Generic model of a business company.

The control loop consists of controlled units like sales, purchase, production and other process areas managed by a regulator unit (the management of the company). The outputs of the controlled units are evaluated by the measuring unit and compared with

¹ Corresponding Author: Roman ŠPERKA, Silesian University in Opava, School of Business Administration in Karviná, Department of Informatics, Univerzitní nám. 1934/3a, 733 40, Karviná, Czech Republic, sperka@opf.slu.cz .

key performance indicators (KPIs). Any differences found are sent to the regulator unit which takes necessary measures in order to run the system in close accordance to the established KPI values. However, it has been shown that a business company must be looked upon as a system with social functions and responsibilities, where individuals act outside of established company KPIs and assert their own personal aims and preferences (see e.g. [9] summarizing the Corporate Social Responsibility research of many other authors). Similarities can be observed in the markets, where customers and suppliers follow their own targets. Further influences such as environment, government decisions, global market fluctuations and other factors also influence the modeled system. Thus, as a result we have to work with rather stochastic system. The previous research results of our approach to this challenge using software agents were presented in [4].

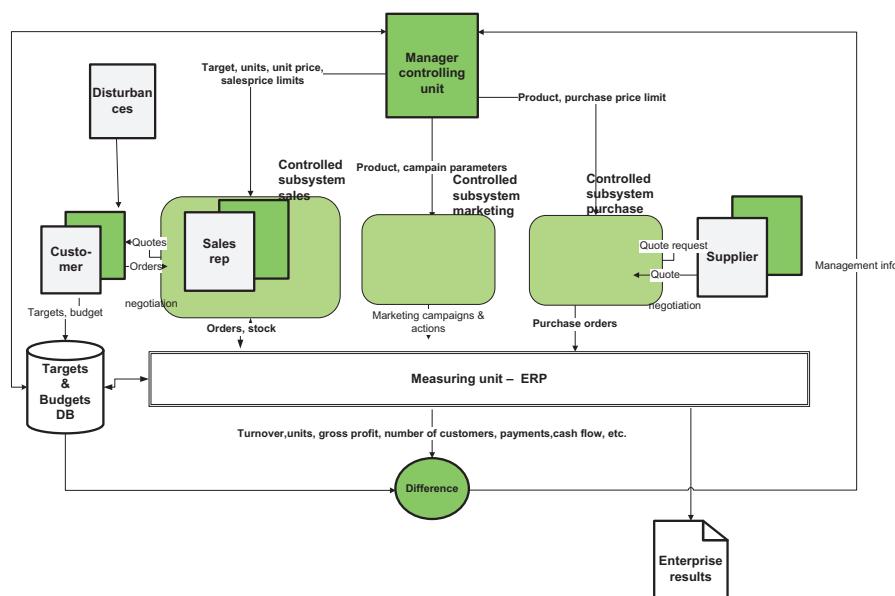


Figure 1. Generic model of a business company (Source: Own).

It is evident that we are dealing with the discrete event simulation using multi-agent paradigm. During the simulation, space and time compression takes place, which helps to save the time and to run a large number of simulation experiments [10-13]. Common business process simulation approaches take advantage of statistical calculations (e.g. [14]). However, several problems can be identified when employing this method. As shown in [11] and mentioned herein above, there are many additional influences that cannot be captured by using a typical business process model (e.g. the effects of the collaboration of business process participants or their communication, experience level, cultural or social factors). Moreover, statistical methods have only limited capabilities of visual presentation while running the simulation. Hence, we do not actually see the participants of a business process dealing with each other during the simulation itself.

One of the problems which business processes simulations struggle with is the lack of real business data. Many researchers [15-17] use randomly generated data instead.

The most useful distributions of random numbers for modeling and simulation are the uniform, normal, exponential, Erlang and Weibull distributions [18]. To obtain useful results from the simulation experiments, it is important to select one of the distributions fitting to the real characteristic of the simulated process. Here must be stated that the fitness of the distribution used may relate to the domain modeled, or more probably to its part.

The motivation of this paper is to research the usability and suitability of different kinds of random number distributions used as input parameters on agent-based simulation outputs. We used the uniform and normal distribution in our simulation experiments, because the uniform distribution seems to model the behavior of customers in the free market like frequency of sales requests while the normal distribution may be related to some characteristics of other model components such as sales representative's ability to sell, customer preferences etc.

This paper is structured as follows: Section 2 briefly informs about the multi-agent system implementation, and negotiation between agents in details. In section 3, the original simulation results are presented.

2. Multi-agent System Implementation

In this section the implementation steps of our multi-agent system and negotiation details are described in detail. Only one part of the company's control loop, defined earlier, was implemented. We suggested the contracts department part of common trading company as the implementation subject. This part consists of sales representatives and customers trading with stock items (e.g. tables, chairs, etc.). One stock item simplification is used in the model simulated. Participants in the contract creation process within our system are represented by the software agents – the sales representative and customer agents interacting throughout the course of the quotation, negotiation, and contracting processes.

During each turn (here we have assumed a week because the normal business negotiation often, or typically needs one week to be completed), the customer agent decides if he wants to buy something. His decision is defined randomly. If the customer decides not to buy anything, his turn is over; otherwise he creates a sales request and sends it to his supplier. The supplier agent answers with a proposal message (concrete quote starting with his maximal price – in our case the maximal price was chosen as *limit price * 1.25*). The reason for this arbitrarily used high first price quoted is a well-known practice in business doing companies by means of which the sales representatives try to achieve a higher than catalogue price. Naturally this approach leads to negotiation, sometimes to complete rejection of the quote in our model.). The agents - customers evaluate the quotes based on “production function” (Eq. 1). The “production function” was proposed to reflect the enterprise market share for the product quoted (market share parameter), sales representatives' ability to negotiate, total market volume for the product quoted etc., in e.g. [4], [19]. If the price quoted is lower than the customer's price obtained as a result of the production function, the quote is accepted. In the opposite case, the customer rejects the quote, and negotiation is started. Sales representative decreases the price to the average of the minimum limit price and current price (in every iteration is getting effectively closer and closer to the minimum limit price) and resends the quote back to the customer. The message exchange repeats until there is an agreement or reserved time passes. The negotiation

between the sales representative (sales representative – agent) is based on the contract net protocol (FIPA contract net protocol [20]). The sales “production function” for m -th sales representative pertaining to i -th customer determines the price that i -th customer accepts.

$$c_n^m = \frac{\tau_n T_n \gamma \rho_m}{Z M \gamma_n^{mi}} \quad (1)$$

c_n^m - the price of the n -th product quoted by m -th sales representative,

τ_n - the company market share for the n -th product $0 < \tau_n \leq 1$,

T_n - the market volume for the n -th product in local currency,

γ - the competition coefficient lowering the sales success $0 < \gamma \leq 1$,

ρ_m - the quality parameter of the m -th sales representative $0,5 < \rho_m \leq 2$,

Z - the number of customers,

M - the number of sales representatives in the company,

y_n^{mi} - the requested number of the n -th product by the i -th customer at m -th sales representative.

Only one stock item is being traded in the simulation and its amount is not limited – the customer will receive as many units as he wants to buy. The sales representative to customer ratio was set at 1:20 – one sales representative serves 20 customers. Customers are grouped in such a way that each group is being served by an assigned sales representative – their relationship is given, neither of them can change their counterpart. Sales process is generally controlled by the manager agent. With each turn, the manager gathers data from all the sales representatives and evaluates the situation of the company. The simulated data collected over a period of time illuminate company behavior – depending on the agents’ decisions and behavior.

Agents in the multi-agent system are situated at two levels. The base agent, ancestor of all Business Process Management (BPM) agents is the “BaseBpmAgent”. It implements base functionality, such as registering to the Yellow Pages, searching for other agents, clean-up and other functions. All the existing agents in the system are descendants of this class. A detailed BPM workflow is described herein under. The customer agent, as in the real market, is the engine of the process. During each turn (one week period) he decides whether he will buy something. If so, he determines the quantity and sends the request to his sales representative. After this the above mentioned negotiation with the sales representative takes place. The result is either the rejection or a sales transaction. Once this negotiation is over, the turn for the specific agent is finished. When all the customer agents finish their negotiations, the turn (one week period) is terminated. In order to arrive to some conclusion, the customer agent needs information about the market – this is where the information agent comes into play. The information agent is responsible for providing information on the market to the customer, but also for the time schedule of the simulation. This agent decides when the turn (one week period) is finished and gives the information to all agents in order to prepare for another turn. After the simulation is over, the “GameOver” information bringing the lifecycle of the agents to the end is broadcasted.

The counterpart - the sales representative agent is in a default semi-sleep state and waiting. Once it receives a request, he immediately replies to the customer agent with the appropriate price and then waits again till the communication is completed. Actually, this agent is not aware of the turn itself. He is informed by the manager agent who requests a report for a one week period. Once per turn the manager agent asks all the sales representative agents about their sales performance. Having received the responses the manager agent evaluates the company situation and creates a report on its current state. In the next section the simulation experiment details will be presented.

3. Simulations

3.1. Setting up the agents

As shown in e.g. [6-8], different number distributions – in our case represented by different pseudorandom generators have significant influence on the result of the simulation. The distributions for simulations in this paper were chosen as follows: customer agent decides whether to buy something or not in particular week of the year using pseudorandom number 1 – 1000 generated from uniform distribution. If this number is higher than 500 then agent decides to ask for CFP, otherwise he does not. Pieces of goods requested by customer agent are generated randomly from the normal distribution. Also the ability of the seller agent is generated from normal distribution. The use of normal distribution here copies the experienced behavior acquired from day-to-day business praxis.

The final generations were performed using “java.util.Random” class for the generation of random numbers. This Java functionality randomly generates values from the uniform distribution. For generating random numbers from the normal distribution, the (Gaussian) Java library, Uncommon Maths, written by Dan Dyer [21] was used. For the generation of more random values, the “MerseneTwisterRNG” class was used. This class is a pure Java port from Makoto Matsumoto and Takuji Nishimura’s proven and ultra-fast Mersenne Twister Pseudo Random Number Generator for C.

3.2. Simulation results

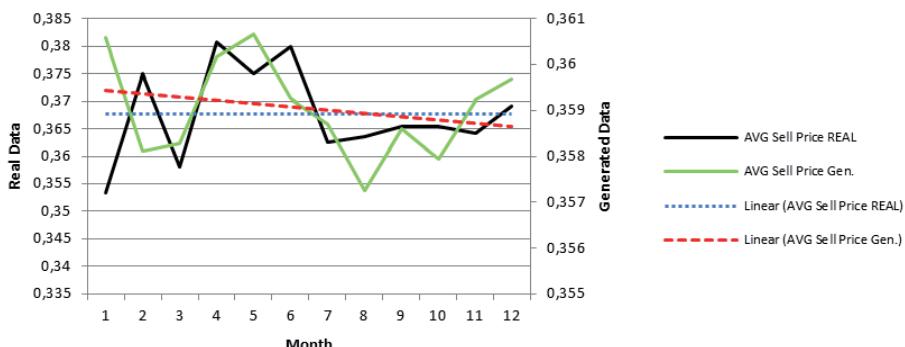
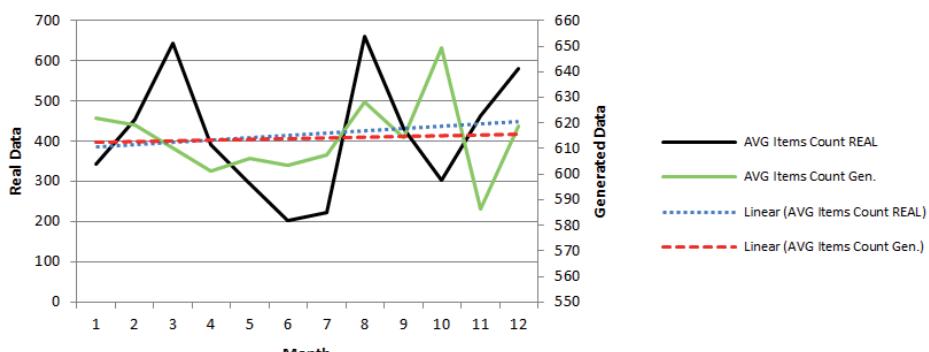
The parameterization of the model and the obtained simulation results are analyzed in this section. Production function (Eq. 1) is the engine of the simulation. Based on it - customer agents decide to buy or not to do so. One year of sales and purchasing processes was simulated. Each turn represents one week. Five simulation experiments were done. Their average values were compared with real data from one computer selling company (30 employees). The real data were taken from an existing company’s accounting information system. Each purchase of the product type was registered. For the comparison of simulated and real data, monthly averages were used. The sales representative agent’s ability (to sell) and the customer purchase quantity attributes were randomly changed in order to model the market fluctuations. The comparison of results obtained by simulation and the real data is presented in the following text.

The parameterization of the MAS is listed in Table 1. This table represents parameters listed by the name and the value for each type of agent (customer, sales representative). It also shows the number of agent type instances (how many of a particular agent types is present in the system).

Table 1. List of agent parameters (Source: Own).

Agent Type	Agent Count	Parameter Name	Parameter Value
Customer	500	Maximum Discussion Turns	10
		Mean Quantity	5
		Quantity Standard Deviation	4
Sales representative	25	Mean Ability	0,7
		Ability Standard Deviation	0,03
		Minimal Price	0,3 €
Manager	1	Purchase Price	0,17 €
Market Info	1	Item Market Share	25 %
		Item Market Volume	1 950 000 €
		Competition coefficient	0,8

The results of the simulation are represented by item price (Figure 2), number of sold items (Figure 3) and the gross profit (Figure 4). All measured time series tend to growing linear trend except for the price. Following graph shows monthly averages of simulated and real price. Real price trend is stable – not growing and not falling. Generated price is quite constant. However, when we depict on the axis with generated data values with 3 decimal places, slightly falling trend is visible. Nevertheless, the real and generated time series have similar development in the time. All curves are located in the range from 0,355 € up to 0,38 €.

**Figure 2.** Prices – month averages (Source: Own).**Figure 3.** Sold items count – month averages (Source: Own).

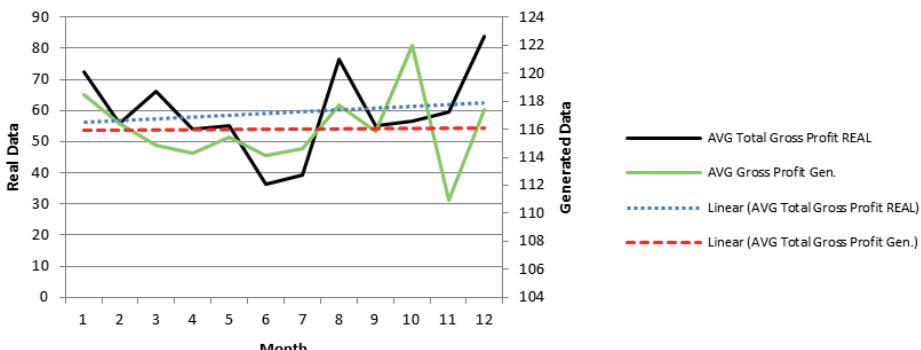


Figure 4. Gross profit – month averages (Source: Own).

From the statistical data listed in the following table it is obvious that the dispersion and standard deviations are higher in real data time series. Real data are more spread in opposite to the generated values. This points to our idea to use the disturbance agent in our simulation framework in order to bring more of unpredictable impacts to the model. The generated values tend to be smoother with lower dispersion. With disturbances we could bring more randomness into the time series development.

Table 2. Statistical data (Source: Own).

	value	average	dispersion	standard dev.
GENER.	Price	0,359	0,000	0,002
	Items Count	613,019	1 988,725	44,595
	Gross Profit	115,879	66,924	8,181
REAL	Price	0,367	0,003	0,055
	Items Count	432,438	720 501,568	848,824
	Gross Profit	61,230	5 200,642	72,115

At this stage of our research we provide only visual comparison of generated and real results. The similar shape of the curves depicted herein above evidently show that the simulation model is valid and we are able to use the method proposed to simulate real business processes properly.

4. Conclusions

The aim of the partial research was to validate, whether business process simulation results show similar outputs in comparison with the real data. Multi-agent system was used as a simulation framework for business process simulations analyzed in this research. Randomized changes to the important internal parameters of software agents used for simulation experiments can lead to similar results even in the case when they are changed by means of pseudorandom functions with different distributions. The results of our simulations show that using a random number generation function with normal distribution leads to correct outputs. Hence, the proposed simulation method could be used for innovative implementation of more complex decision support tool.

Using other types of distribution as well as the implementation of an entire company's control loop, and validation with more complex statistical tests will be the task and subject of the research in the near future.

References

- [1] P. Suchanek, D. Vymetal, Security and Disturbances in e-Commerce Systems, In: Proceedings of the *10th International Conference Liberec Economic Forum 2011*. (2011), ISBN 978-80-7372-755-0.
- [2] Y. Liu, K.S. Trivedi, *Survivability Quantification: The Analytical Modeling Approach*, Department of Electrical and Computer Engineering, Duke University, Durham, NC, U.S.A., 2011, <http://people.ee.duke.edu/~kst/surv/IoJP.pdf>. Accessed 16 January 2012.
- [3] M. Barnett, *Modeling & Simulation in Business Process Management*, Gensym Corporation, pp. 6-7. Available from: <http://news.bptrends.com/publicationfiles/11-03%20WP%20Mod%20Simulation%20of%20BPM%20-%20Barnett-1.pdf>, 2003. Accessed 16 January 2012.
- [4] D. Vymetal, R. Sperka, Agent-based Simulation in Decision Support Systems, *Distance learning, simulation and communication* (2011). Proceedings. ISBN 978-80-7231-695-3.
- [5] P. Wolf, *Úspěšný podnik na globálním trhu*. Bratislava: CS Profi-Public.2006, ISBN 80-969546-5-2.
- [6] R. Šperka, M. Spišák, K. Slaninová, J. Martinovič, P. Draždilová, Control Loop Model of Virtual Company in BPM Simulation, In: Proc. *Advances in Intelligent Systems and Computing* (2012). Soft Computing Models in Industrial and Environmental Applications. Berlin Heidelberg: Springer-Verlag, Germany, Volume 188, pp. 515-524. DOI: 10.1007/978-3-642-32922-7_53. ISSN 2194-5357. ISBN 978-3-642-32921-0.
- [7] K. Slaninová, J. Martinovič, P. Draždilová, D. Vymětal, R. Šperka, Analysis of Agents' Behavior in Multiagent System, In: Proc. 24th European Modeling and Simulation Symposium. EMSS 2012, Vienna, Austria. 19.-21.9.2012. Rende: Universita di Genova, Italy, pp.169-175. ISBN 978-88-97999-09-6.
- [8] D. Vymětal, M. Spišák, R. Šperka, An Influence of Random Number Generation Function to Multiagent Systems, In: Proc. LNAI 7327. *Agent and Multi-Agent Systems. Technologies and Applications* (2012), Berlin Heidelberg: Springer-Verlag, Germany, pp.340-349. ISSN 0302-9743. ISBN 978-3-642-30946-5. DOI 10.1007/978-3-642-30946-5.
- [9] S. Sharma, J. Sharma, A. Devi, Corporate Social Responsibility: The Key Role of Human Resource Management, *Business Intelligence Journal* (2009).
- [10] C.M. Macal, J.N. North, Tutorial on Agent-based Modeling and Simulation, In Proceedings: *Winter Simulation Conference* (2005).
- [11] M. Sierhuis, *Modeling and Simulating Work Practice*, PhD thesis, University of Amsterdam, 2001.
- [12] C. De Snoo, *Modeling planning processes with TALMOD*, Master's thesis, University of Groningen, 2005.
- [13] N.R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, B. Odgers, Autonomous agents for business process management, *Int. Journal of Applied Artificial Intelligence* **14** (2000), pp. 145–189.
- [14] A.W. Scheer, M. Nuttgens, *ARIS architecture and reference models for business process management*, Bus. In: van der Aalst WMP, Diesel J, Oberweis A (eds.) *Business Process Management*. LNCS, vol. 1806, pp. 376–389. Springer, Heidelberg, 2000.
- [15] B. Dunweg, W. Paul, *Int. J. Mod. Phys. C* **2**, pp. 817, 1991.
- [16] J. Hillston, *Random Variables and Simulation*. Available from: <http://www.inf.ed.ac.uk/teaching/courses/ms/notes/note13.pdf>, 2003. Accessed 16 January 2012.
- [17] H.C. Ottinger, *Stochastic Processes in Polymeric Fluids*. Springer Verlag, 1996.
- [18] J. Noble, *Computational Modeling*, Lecture notes and practical exercises. University of Leeds. Available from: <http://users.ecs.soton.ac.uk/jn2/simulation/>, 2011. Accessed 16 January 2012.
- [19] D. Vymetal, C. Scheller, MAREA: Multi-Agent REA-Based Business Process Simulation, *ICT for Competitiveness* (2012). Proceedings. Karviná, 320 p, pp. 300 – 310. ISBN 978-80-7248-731-8.
- [20] Foundation for Intelligent Physical Agents (FIPA), *FIPA Contract Net Interaction Protocol*, In Specification [online]. FIPA, Available from: <http://www.fipa.org/specs/fipa00029/SC00029H.pdf>, 2002. Accessed 13 June 2012.
- [21] D.W. Dyer, *Uncommons Maths - Random number generators, probability distributions, combinatorics and statistics for Java*. Available from: <http://maths.uncommons.org/>, 2010.

The Supply Chain in Cloud Computing – the Natural Future

Katarzyna GRZYBOWSKA^{a,1}, Gábor KOVÁCS^b, and Balázs LÉNÁRT^b

^a*Poznan University of Technology, Faculty of Engineering Management, Poland*

^b*Budapest University of Technology and Economics, Faculty of Transportation
Engineering and Vehicle Engineering, Hungary*

Abstract. Firstly, the paper presents the definition and evolution of supply chains. On the other hand, it details the use of mobile solutions in the supply chain and the characteristics, models and forms of cloud computing. Based on these, the paper describes the IT technical approach of cloud supply chains, and it presents the electronic freight and warehouse exchanges as a type of hybrid cloud supply chains. Moreover, the paper contains the new challenges and opportunities of cloud supply chains.

Keywords. Cloud Computing, Supply Chain, Cloud Supply Chain, Online Logistics Exchanges

1. Introduction

The Supply Chain (SC) (also referred to as the net of connections) is nothing but an expanded system within each link (enterprise) constitutes a separate subsystem. In [1], [2] and [3] there is defined the Supply Chain as a network of places. Supply chain – a set of three or more companies directly linked by one or more of the upstream and down-stream flows of products, services, finances, and information from a source to a customer [4]. A Supply Chain is the alignment of firms that bring products or services to market [5]. Supply Chain Management is “the systemic, strategic coordination of the traditional business functions and the tactic across these business functions within a particular company and across businesses within the Supply Chain, for the purposes of improving the long-term performance of the individual companies and the Supply Chain as a whole” [6]. The Supply Chain is a metastructure. The metastructure is an intermediate form between a single enterprise (microstructure) and global economy [7]. It is understandable, since the enterprises do not exist in isolation from the environment in which they function: the organizations do not act in isolation and their survival is often dependent upon effective interactions with the environment [8]. This integration should be based on close and partner cooperation [9].

The systemic understanding of the enterprises operating in the Supply Chain is also the starting point and the basis for designing effective Supply Chain which would be characterized by more effective competitiveness and meeting of recipients’ and

¹ Corresponding Author: Katarzyna Grzybowska, Poznan University of Technology, Faculty of Engineering Management, Strzelecka 11, 60-965 Poznań, Poland; E-mail: katarzyna.grzybowska@put.poznan.pl

consumers' expectations. The systemic approach is the skill to see the problem as a whole as well as the relations that connect individual elements (enterprises) thereof and the permanence of changes that take place over time. Supply Chain Management (SCM) has become an important management paradigm [10].

One may characterize three specific stages of Supply Chain development from the functional, through the reactive to those of an adaptive nature. This development is, *inter alia*, the result of change of relationships (dependencies and connections) that take place between the enterprises in the Supply Chain. Key characteristics and relationships in the Supply Chain are shown on Table 1.

Table 1. The evolution of Supply Chain [12]

Attribute	Functional Supply Chain to the 1980s	Responsive Supply Chain the 1990s	Adaptive Supply Chain the 2000s
Integration focus	Over the wall Reactive/quick fixes Monopoly suppliers	Transactional Responsive Competition is suppliers	Collaboration Decision/proactive Joined-up networks of enterprises
Customer focus	Customer can wait 'you will get it when we can send it'	Customer wants it soon 'you will have it when you want it'	Customer wants it now you will get it'
Organisation focus	Departmental and ring fencing	Intra-enterprise 'internal' involvement	Extended enterprise involvement
Product positioning	Make to stock Decentralized stock holding Store then deliver	Assemble to order Centralised stock holding Collect and cross dock	Make to order Minimal stock holding Whatever is needed
Management approach	Hierarchical	Command and control	Collaborative
Technology focus	Point solution	ERP	Web connected
Time focus for the business	Weeks to months	Days to weeks	Real time
Performance focus	Cost	Cost and service	Revenue and profit
Collaboration	Low	Medium	High levels
Response time	Static	Medium	Dynamic

In functional Supply Chains, repeatable and routine operations are carried out. The approach towards business partners is antagonistic and changing. Business partners cooperating in the Supply Chain apply the following rule: if I win, you have to lose. The risk of enterprise and cooperation is transferred to the business partner. As a result of all the above, the cooperation is not lasting long.

The reactivity of the links composing the Supply Chain is interpreted as a relatively constant intensity of reacting to external signals (resulting from the environment in which the Supply Chains functions) and internal signals which come from other links of the Supply Chain and from the enterprise itself. The enterprises thus function on the basis of action-reaction rule. They accommodate their own as well as mutual needs and expectations. These actions are both of reactive nature, forced by the business partner and adaptive, carried out at own initiative of the enterprise. As a result of reactivity of the Supply Chain, a relative synergy of joint actions of enterprises takes place. Considerable support and facility for maintaining the reactivity of a Supply Chain is implemented in ERP-class IT systems and in other telecommunication technologies. The available resources of cooperating enterprises are appointed to specific goals. Selection of both the resources and business partners is oriented on competitiveness goals set out [11].

A mutual interdependency and cooperation on strategic level occurs between the enterprises. Business partners are allowed to cooperate in creating joint strategy of the supply chain. The nature of the development of adaptive SCs are to make joint decisions based on partnership principles, taking of actions that serve integration of

enterprises as well as integrating processes carried out within the scope of the Supply Chain. The goal thereof is to increase efficiency of flow in the Supply Chain and improvement of competitiveness of all participants of the Supply Chain. Adaptive Supply Chain networks possess the flexibility to continually morph and respond to the environment in near real time without compromising on operational and financial efficiencies. These networks seamlessly connect supply, planning, manufacturing, and distribution operations to critical enterprise applications and provide near real-time visibility across the supply network, thereby enabling rapid decision making and optimal execution [13].

In order to make Adaptive Supply Chain even more efficient and adaptive to market requirements, the authors propose a new solution based on Cloud Computing. In cloud computing, the user's computer may contain almost no software or data (only an operating system and a web browser). The provider's cloud computing services form the cloud. These services are provided via an Internet connection within one or more of the next layers: application, platform and infrastructure [14]. The application of the Supply Chain concept in the context of cloud computing is innovative and opens a new research field [15]. Lindner et al. present definition of Cloud Supply Chain (CSC). It is two or more parties linked by the provision of cloud services, related information and funds. The Supply Chain represents a network of interconnected businesses in the cloud computing area.

Figure 1 is part of an exploration on moving from the old methods of running our businesses (in the 20th century) – including business models, processes, and technologies – to the new (the 21st century) [16]. Characteristics of Cloud Supply Chain:

- Primary goal: Supply Chain demand at the lowest level of costs and respond quickly to demand,
- Product design strategy: Create modularity to allow individual setting while maximizing the performance of services,
- Pricing strategy: Lower margins, as high competition an comparable products,
- Manufacturing strategy: High utilization while flexible reaction on demand,
- Inventory strategy: Optimize of buffer for unpredicted demand, and best utilization,
- Lead time strategy: Strong Service Level Agreement (SLA) for ad-hoc provision,
- Supplier strategy: Select on complex optimum speed, cost, and flexibility,
- Transportation strategy: Implement highly responsive and low cost modes.

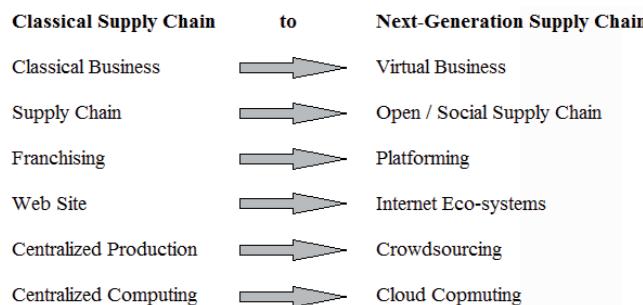


Figure 1. Moving from Classical to Next-Generation: Cloud Supply Chain [16].

2. Migration of Supply Chains to the Cloud

Mobile SCM (mSCM) integrates software applications with mobile devices (e.g. cell phones, personal digital assistants, tablets, on-board computers, see Figure 2) to give users the flexibility to operate in a wireless environment at any location. Mobile devices are connected to the company's computer server via wireless technology infrastructure such as GSM/3G, Wi-Fi capable equipment or through modern satellite providers (e.g. Inmarsat BGAN). These enable users to share data across functions and along the supply chain without the need for fixed wired connections for exchange of real time information. In the early stages of mobile communication short message service (SMS) software applications were used to access company databases, nowadays service oriented architecture (SOA) is preferred with standardized XML (Extensible Markup Language) communication [17].

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. According to NIST specification (National Institute of Standards and Technology, U.S. Department of Commerce) cloud model is composed of three service models (Software as a Service /SaaS/, Platform as a Service /PaaS/, Infrastructure as a Service /IaaS/), and four deployment models (Private-, Community-, Public- and Hybrid Cloud), [18]. Figure 2 shows the overview of cloud computing.

Before migrating enterprise Supply Chain to the cloud numerous assumptions must be analyzed. Implementation of new technologies requires high level IT culture and developed IT infrastructure. As cloud computing is a new area of IT technologies preparation is needed in key areas such as standardization technology, virtualization technology, data management technology, platform management technology in supply chain information collaboration [19].

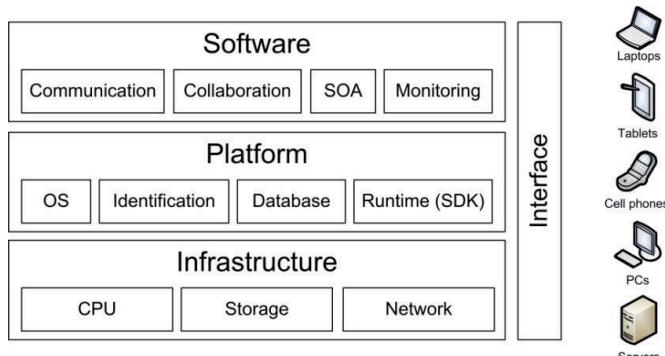


Figure 2. Diagram showing overview of cloud computing.

Regarding to the dynamic changing computational and storage demand the Cloud computing service providers (e.g. Amazon EC2, Microsoft Azure) use virtualization technologies to maintain resources mainly for load balancing and management purposes. Hardware virtualization refers to the creation of virtual appliances such as computers, storage and network devices. In this case a hypervisor is installed on the physical hardware. The hypervisor runs the virtual entities and manages the hardware resources such as CPUs, memory, hard disks and I/O devices.

Cloud service providers provide classical data storage and database as virtual machine image (e.g. Oracle Database 11g Enterprise Edition) or as a service (e.g. Amazon Relational Database Service or Microsoft SQL Azure) within the cloud. The advantages of these services are scalability, high availability, reliability compared to the traditional solutions. These factors are highly important as the key element of the Supply Chain Management is information. With high quality services and reliable data communication techniques this goal can be achieved.

One of the most challenging questions in Supply Chain Management is data exchange between service providers. The traditional ways of changing data is Electronic Data Interchange (EDI), which is a structured electronic data transmission standard between computer systems. This standard was developed in 1996 and it is capable of computer-to-computer communication over computer network (e.g. VAN, BBS). As it was developed for point-to-point communication the current state of EDI is not able to grant the present need of business-agile enterprises. Collaboration of companies nowadays is a key factor and the dynamic interchange of information is a need.

Along with the development of information technology, internet data transmission became secure, scalable and standard for data communication. Compared to the EDI internet transmission cost is far lower, the enterprise only need to open a SOA based web service without having to acquire additional equipment and increase the professional management staff. In the supply chain management the mode of information collaboration and information service object is also constantly changing, the supply chain members may quit and also new members may join at any time.

As mentioned before collaboration and data exchange between companies within the whole supply chain is a common need in logistics. Such as traditional point-to-point EDI is no more acceptable for the market. The existing ERP and CRM systems are capable for the communication through its standards (e.g. SOA), developed in early 2000s, but a centralized solution not exists on the market, which can reduce supply chain information distortion, accelerate information transmission speed and accuracy, and improve the overall competitiveness of supply chain's role. With an integrated cloud service application such as electronic freight and warehouse exchange a controlled collaboration can be achieved within the whole supply chain (Section 3).

3. A Hybrid Cloud Supply Chain Model: Electronic Freight and Warehouse Exchange

The electronic freight and warehouse exchange facilitates a forum for logistics service providers to advertise their service supply, such as transport and storage on the worldwide web; whereas customers can choose the offer, which best suits their needs. The cloud structure of the electronic freight and warehouse exchange is shown by Figure 3 [20]. The freight and warehouse exchange (cloud computing service provider) offers the following main services [21]:

- e-commerce toolbar (agile information and communication techniques):
 - (1) advertising and searching in a simple catalogue (freight/storage tasks/capacities),
 - (2) automatic offer sending (based on individual settings),
 - (3) tenders/auctions (just for freight/warehouse tasks),
- multi-criteria decision supporting algorithms (choose the best offer),
- optimization algorithms (optimize the logistics processes),

- other functions (e.g. statistics, blacklists, data maintenance, etc.).

The electronic freight and warehouse exchange has three participants: consigners, logistics providers, and the cloud computing service provider. The aims of the electronic freight and warehouse exchange: to advertise freight/storage capacities/tasks, to choose suitable offers based on e-commerce methods and complex optimum criteria, to support complex logistics processes (e.g. combined transport, city logistics, etc.).

In these exchanges there are lots of optimization opportunities, e.g. in case of freight and warehouse exchanges, we have to define a complex objective function. On a part of the total transport route, the freight tasks are transmitted together and then with the help of a combi terminal the freight tasks are transferred (multimodal transportation with rail/river) [22]. These problems can be solved by ACO (ant colony optimization), which is an optimizing algorithm developed by Marco Dorigo [23] based on the modeling of the ants' social behavior. In the electronic freight and warehouse exchange similar problem emerges as the ants' search for food: the target is the agile performance of freight/storage tasks offering the higher profit.

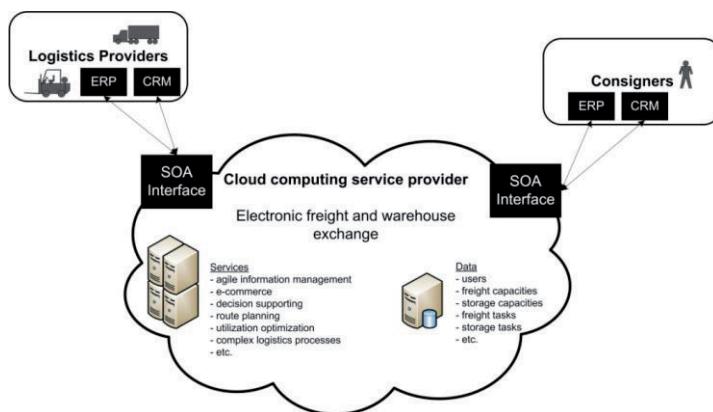


Figure 3. The cloud model of electronic freight and warehouse exchanges.

The role of freight and warehouse exchanges in complex logistics problems (city logistics, combined transportation) may be viewed as the route planning systems of companies (e.g. wholesalers): the processes (e.g. tours, utilization) can be optimized by handling demands and capacities in one system. Moreover, through the coordination they are able to establish collecting-distributing routes, to organize back haul, and through this to reduce the number of vehicles. In this way, support of complex logistics problems (city logistics, combined transportation) will be possible. In other words, freight and warehouse exchanges is one of the “simplest”, but still the most efficient way of optimizing complex logistics processes. Moreover, the freight and warehouse exchanges can be an excellent example of cloud supply chains.

4. New Challenges Cloud Supply Chain – the Basis for Creating a New Existence Next-Generation

Modern times (globalization, specialization and the use of World Wide Web) – create the hybrid SCM 2.0 (Supply Chain Management 2.0). This is the subsequent development

phase of a Supply Chain intended to increase its effectiveness in the exchange of information and cooperation between the participants of a Supply Chain [24].

The Cloud Supply Chain offers: (1) limitless flexibility: With access to millions of different pieces of software and databases, and the ability to combine them into customized services, users are better able to find the answers they need, share their ideas; (2) better reliability and security; (3) enhanced collaboration: By enabling online sharing of information and applications, the Cloud offers users new ways of working together; (4) portability: Users can access their data and tools wherever they can connect to the Internet; (5) simpler devices: With data and the software being stored in the Cloud, users do not need a powerful computer. They can interface using a cell phone, Personal Digital Assistant [25] and (6) synchronization: The Cloud offer tools enable business units and multiple plants across an enterprise and SC to communicate with one another more effectively. Authors identified the barriers to adoption of SC in Cloud Computing; (7) standardization: So far, there are no clearly defined and widely adopted standards, though this would be beneficial to cloud computing customers and service developers [26]. Standardization refers to the use of common APIs (Application Programming Interface) and architectures, as well as, technical standards (ANSI – American National Standards Institute or the ISO – International Organization for Standardization, or simply a commonly-used or familiar interface). The elements of the system are identified which are relevant to the problem:

- Cooperation and partner eco-system: The concept of an eco-system in the natural world hasn't changed. What has changed is our understanding of the potential that such a concept represents when applied to traditional business concepts of partnering. A partner eco-system's depends upon interdependence – it is a highly-collaborative and complexly-orchestrated. Cooperation and partner eco-system applied as parts of a business strategy, it creates a partnering environment where group strength drives improvements far greater than those that can be realized by any individual organization.
- Sharing information: The major challenge of the Cloud Supply Chain coordination is to find a global solution for the composite service. Therefore, we need to coordinate the flow of information and link the business processes under various constraints. The Cloud Supply Chain coordination leads to take the advantage of the web services standardized communication protocols. This simplifies communication among the firms in the Supply Chain, and thus facilitates collaboration [27].
- Trust: Trust is a factor of the relationships. As pointed out by R. Hardin trust is a characteristic of interest [28]. Thus, trust means [29]: (1) trusting others and assuming their reliability (unless proved otherwise), (2) treating trust put in oneself seriously and meeting other parties expectations (unless the trust proves unauthentic).

In a Supply Chain, trust is one of the key cooperation factors (e.g. trust that a supplier or a sub-contractor perform their duties according to specifications; trust that a supplier with which the enterprise did cooperate previously, will supply a product of a proper quality; trust that the customer will pay within agreed period of time and will not cause a payment gridlock, etc.). In such cases, trust is a mandatory condition of cooperation – it is like a grease that facilitates the cooperation of elements [30]. New developments include the offering of computer-business that are completely hosted in the Cloud. This will make portability easier, as the software can be resumed from a

different location [31]. Also, it is less dependent on the user's hardware and less prone to piracy.

The effective management of the SC according to [32] and [33] is based on the creation of a virtual organization by combining a number of commercial entities (for example the Cloud Supply Chain). These, in turn, supplement each other in pursuing a joint objective. In order to ensure the versatile success of the created alliances, it is purposeful that the Cloud Supply Chain be based on trust and the involvement of all parties of the Supply Chain.

5. Conclusion

The main advantage of the freight and warehouse system is, that a manifold optimum search tool is available in the electronic freight and warehouse exchanges. With the help of the mentioned methods by the filtering of local optimums, a solution can be found shortly, which to freight/storage capacities/tasks selects freight/storage capacities/tasks. There are a lot of optimization opportunities, from the decision making, to the route planning and utilization optimising. In addition, complex e-commerce methods (e.g. tender, auction) help the selection.

The role of freight and warehouse exchanges in complex logistics problems (city logistics, combined transportation) may be viewed as the route planning systems of companies (e.g. wholesalers): the processes (e.g. tours, utilization) can be optimized by handling demands and capacities in one system. Moreover, through the coordination they are able to establish collecting-distributing routes, to organize back haul, and through this to reduce the number of vehicles. In this way, support of complex logistics problems (city logistics, combined transportation) will be possible. In other words, freight and warehouse exchanges is one of the "simplest", but still the most efficient way of optimizing complex logistics processes. Moreover, the freight and warehouse exchanges can be an excellent example of Cloud Supply Chains.

References

- [1] R. Ganeshan, T.P. Harrison, *An Introduction to Supply Chain Management*, Penn State University, University Park, PA., 1995.
- [2] D. Brown, S. Wilson, *The Black Book of Outsourcing* Hoboken, Wiley, New Jersey, 2005.
- [3] H.L. Lee, C. Billington, The Evolution of Supply-Chain-Management Models and Practice at Hewlett-Packard, *Interfaces* **25** (1995), 42-63.
- [4] J.T. Mentzer, *Supply Chain Management*, Sage Publications Ins, 2001.
- [5] D.M. Lambert, J.R. Stock, M. Ellraml, *Fundaments of Logistics Management*, Irwin/McGraw-Hill, Boston, 1998.
- [6] J.T. Mentzer, W. Dewitt, J.S. Keebler, S. Min, N. Nix, C.D. Smith, G. Zachariaz, Defining Supply Chain Management, *Journal of Business Logistics*, **22** (2001), 1-25.
- [7] K. Grzybowska, Consistent supply chain – a way to increase the efficiency of metastructures and survival in a crisis, *Change in condition for success* [in Polish], **128** (2010), 319-326.
- [8] S.P. Robbins, D.A. Decenzo, *Fundamentals of Management*, Pearson Education. New Delhi, 2001.
- [9] G. Nizard, *Metamorphoses of a Company: Management in a Changing Organization's Environment*, [in Polish], PWN, Warszawa, 1991.
- [10] H.L. Lee, V. Padmanbhan, S. Whang, The Bullwhip Effect in Supply Chains, *Sloan Management Review*, **38** (1997), 93-102.
- [11] K. Grzybowska, Change of relationships within the supply chain – creating the culture of cooperation, (Grzybowska K., Stachowiak A., Eds.), *Integration of supply chains – modeling, partnership and controlling*, Publishing House of Poznan University of Technology, Poznan, 2009.

- [12] S. Emmett, B. Crocker, *The relationship-driven supply chain: creating a culture of collaboration throughout the Chain*, Gower Publishing Ltd., Hampshire, 2006.
- [13] SAP: *Adaptive Supply Chain Networks*, 2002.
- [14] D.A. Marincas, C. Voicila, Using Web Technologies for Supply Chain Management, (Önkal D., Aktas E., Eds.), *Supply Chain Management – Pathways for Research and Practice*, InTech, Rijeka, 2011.
- [15] M. Lindner, F. Galan, C. Chapman, S. Calyman, D. Henriksson, E. Elmroth, *The Cloud Supply Chain: A Framework for Information, Monitoring, Accounting and Billing*, Springer Verlag, 2010.
- [16] D. Hincliff, <http://www.flickr.com>, 2011.
- [17] Teck-Yong, Mobile supply chain management: Challenges for implementation, *Technovation*, **26** (2006), 682-686.
- [18] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*. National Institute of Standards and Technology, U.S. Department of Commerce, 2011.
- [19] J. Chen, W. Ma Yan, The Research of Supply Chain Information Collaboration Based on Cloud Computing, *Procedia Environmental Sciences*, **10** (2011), 875-880.
- [20] G. Kovács, The structure, modules, services and operational process of modern electronic freight and warehouse exchanges, *Periodica Polytechnica Transportation Engineering*, **37** (2009), 33-38.
- [21] G. Kovács, Possible methods of application of electronic freight and warehouse exchanges in solving the city logistics problems. *Periodica Polytechnica Transportation Engineering*, **38** (2010), 25-28.
- [22] G. Kovács, K. Grzybowska, Logistics processes supported by freight and warehouse exchanges. Grzybowska, K., Golińska, P. (Eds.) *Selected logistics problems and solutions*, Publishing House of Poznan University of Technology, Poznan, 2011.
- [23] M. Dorigo, T. Stützle, *Ant colony optimization*, MIT Press, Cambridge, 2004.
- [24] K. Grzybowska, Creating trust in the supply chain, (Grzybowska K., Ed.), *New insights into supply chain*, Publishing House of Poznan University of Technology, Poznan, 2010.
- [25] A. Cavoukian, Privacy in the clouds, *Identity in the Information Society*, **1** (2008), 89-108.
- [26] C.N. Höfer, G. Karagiannis, Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*, **2** (2011), 81-94.
- [27] I. Mahdavi, S. Mohebbi, M. Zandakbari, N. Cho, N. Mahdavi-Amiri, Agent-based web service for the design of a dynamic coordination mechanism in supply networks, *Journal of Intelligent Manufacturing*, **20** (2009), 727-774.
- [28] R. Hardin, Trusting persons, trusting institutions, (Zeckhauser R. J., Ed.), *Strategy and Choice*, MIT Press Cambridge, 1991.
- [29] P. Sztmporka, *Trust: the Foundation of Society*. [in Polish]. Wydawnictwo Znak, Kraków, 2007.
- [30] P. Dasgupta, Trust as a commodity, (Gambetta D., Ed.), *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell Oxford, 1988.
- [31] C. Edwards, *The Tech Beat: games in the cloud?*, http://www.businessweek.com/the_thread/techbeat/archives/2009/02/games_in_the_cl.html, 2009.
- [32] T. Lehtonen, Attributes and success factors of partnering relations—A theoretical framework for facilities services, *Nordic Journal of Surveying and Real Estate Research—Special Series*, **2** (2004), 31-46.
- [33] I. Chen, A. Paulraj, Understanding supply chain management: critical research and a theoretical framework, *International Journal of Production Research*, **42** (2004), 131-163.

Incentive Contracts in Logistics Outsourcing

Qin ZHU, and Richard Y.K. FUNG¹

System Engineering and Engineering Management, City University of Hong Kong

Abstract. Recent development in logistics has led to wide use of external service providers. This paper aims to apply the agency theory to study the cooperative relationship between outsourcing enterprise and their logistics service providers through the use of incentive contracts. The problem is modeled as a risk-neutral outsourcing enterprise hiring multiple risk-averse service providers to conduct different logistics tasks. The outsourcing enterprise is regarded as principal, while their service providers are referred as agents. Each agent makes his own decision on the amount of additional work to exert into the project. The incentive provided by the principal is typically tied to the amount of time ahead of scheduled deadline. The goal of this paper is to study the design of optimal contracts that balance the incentives and risks of the agents from the view of the principal, in order to achieve maximized profits for all the parties.

Keywords. logistics outsourcing, incentive contracts, agent theory, risk aversion

1. Introduction

Logistics outsourcing has been rapidly growing during the past two decades. Intense competition in the global supply chain environment has caused many organizations to focus on their core competences and outsource certain business processes to external LSPs (logistics service providers). These LSPs are more professional in specific business process. They help to reduce operational cost and improve productivity, providing organizations the feasibility to concentrate on crucial business process [1, 2].

Although logistics outsourcing has brought a lot of benefits to the supply chain, it also causes problems. The major problem in logistics outsourcing lies in the possible moral hazard [3]. The source of moral hazard is from information asymmetry, such that both the principal and the agent do not share the private information with each other. In this paper, incentives from the principal motivate the agents to exert more additional work to accelerate their working processes, which would increase their corresponding costs and risks. However the cost structures and degree of risk aversion of the agents are private information to the principal. Agency theory can be exactly applied to solve this problem, by offering the agent a contract that links his compensation to the output [4-6].

Incentive contracts provide the framework for analyzing the strategic interactions among principal and agents in the traditional agency problems. The spirit of incentive

¹ Corresponding Author: Richard Y.K. FUNG, System Engineering and Engineering Management, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong; E-mail: meykfung@cityu.edu.hk.

contract is to find theoretical ways to motivate agents to take appropriate actions, to realize the goal of the principal. Incentive contracts serve the dual function of rewarding productive efforts and allocating risks [7]. Incentive contracts have been widely used in supply chain coordination [8, 9]. They are believed to allow two main objectives: one is to increase total supply chain profits by aligning objectives of all supply chain members; the second is to share the risks [5].

This paper contributes to the multi-task principal-agent literature by including a time-cost trade-off. As in outsourcing project, plenty of enterprises are involved, multi-task principal-agent model is used. Each agent is responsible for a specific task, and completion of the whole project depends on the realization of each task. Reasons for including time-based incentives are vast, for example, early completion could insure rapid return on investment, and project delays could cause serious sequel such as loss of customers' orders. As a result, time-based incentive sets a reward for each day of early completion. Different from traditional time-based incentive contracts, costs of risks are also considered in this paper, by assigning each agent a degree of risk aversion. Risk aversion captures the reluctance of the agent to accept a bargain with an uncertain payoff rather than another more certain but possibly lower payoff. Many researchers ignore the risk aversions of the supply chain members, which may affect the achievability of channel coordination. The present study assumes that all the agents are risk-averse while the principal is risk-neutral for two reasons. First, the cost of bearing risk is generally relative less for the outsourcing enterprise than for the LSPs. Second, it conforms to the research tradition on the agency literature.

2. Basic Model

The basic scenario is as follows: an outsourcing enterprise receives an order from his customer that consists of N tasks (for example, transportation, customs declaration, delivering, and so on). He hires N external service providers to conduct these logistics functions, and each service provider is responsible for one task. The situation is that all the tasks should be performed in sequence (compared to the parallel case, where all the tasks are conducted in the same time). For example, service provider A transports the products from the manufacturing factory to the local warehouse firstly, and then passes the work to service provider B for the customs declaration process, afterwards service provider C is responsible for delivering the products to retails. The outsourcing enterprise will receive payment from the customer at the end of the project, depending on the whole realized project completion time. The task completion time of each service provider depends on how much additional work it will exert to finish the job ahead of scheduled deadline. The compensation from the outsourcing enterprise to each service provider is composed of two parts: a fixed payment and a bonus. The bonus is settled as a reward for each day of early task completion. The project finishes only when all the tasks are completed.

2.1. Notations and Assumptions

The realized task duration D_i for each service provider is the scheduled task duration minus the amount of time moved up, in the form that:

$$D_i = d_i - A_i + \varepsilon_i, A_i \leq D_i \quad (1)$$

where d_i is the scheduled task duration for the task i . A_i is the time that the service provider i saved before deadline, which may be achieved through additional work, such as extending working hours, introducing more experienced people, or building up more advanced equipment. ε_i represents the external environmental uncertainty that the service provider cannot control, such as the weather condition. The environmental uncertainty has a non-negligible effect on the realized task completion time. It is assumed that the stochastic terms ε_i follows normal distribution, with a zero mean and fixed variance of σ^2 . The related cost of additional work of service provider i is:

$$V_i(A_i) = c_i A_i^2 / 2 \quad (2)$$

where c_i is the cost coefficient for additional work conducted by the i -th provider. The cost of additional work $V_i(A_i)$ is assumed to be convexly increasing, such that $V'_i(A_i) \geq 0$, and $V''_i(A_i) \geq 0$. The revenue of the outsourcing enterprise, R_{elp} , is defined as follows:

$$R_{elp} = r_0 - \alpha D_{tot} \quad (3)$$

where r_0 and α are positive values. r_0 captures the base price the enterpriser receives upon completion of the project. α represents the incentive for early completion. Owing to the serial case, the total project duration D_{tot} are simply the sum of individual task completion time D_i , that is $D_{tot} = \sum_{i=1}^n D_i$.

The compensation scheme S_i is composed of a fixed fee, s_{0i} , and a bonus, decreasing with the real task completion time D_i :

$$S_i = s_{0i} - \beta_i D_i \quad (4)$$

where β_i is defined as the incentive intensity decided by the enterprise. It is the key decision variable of the principal. Substituting Eq. (1) into Eq. (4), so that the expected value of the compensation $E[S_i]$ is $E[S_i] = s_{0i} - \beta_i d_i + \beta_i A_i$, and the corresponding variance is $Var[S_i] = \beta^2 \sigma^2$.

The utility function of the agent is assumed to be negative exponential, that is: $U(\Pi_i) = -\exp(-k_i \Pi_i)$, where k_i is the coefficient of the risk aversion of the agent. The profit of service provider Π_i , is calculated by subtracting the associated cost from the compensation given by the principal. The important feature of this utility function is the constant absolute risk aversion. This form insures that the wealth of the agent does not affect his risk aversion and therefore does not affect the incentives from the principal. Then the expected utility of the agent i can be expressed in terms of Certainty Equivalence CE_i , such as:

$$CE_i = E[U(\Pi_i)] = E[S_i] - k_i VAR[S_i]/2 - V_i(A_i) \quad (5)$$

2.2. First-Best Solution

As a benchmark for incentive contracts literature, agency theory papers generally firstly compute the solution to the problem assuming away the moral hazard problem, called “first-best solution”. In this situation, the cost information of the agent is public to the principal. There is only one constraint: incentives offered by the enterprise should be more attractive than the outside options h_i in order to motivate the agents to involve in the project. Due to the properties of the serial task case, the whole problem can be decomposed into N single agent problems:

$$\max_{s_i} P_i^{FB} = E[R_i] - E[S_i] \quad (6)$$

$$\text{s.t. } E[S_i] - k_i Var[S_i]/2 - V_i(A_i) \geq h_i \quad (7)$$

$$P_{ep}^{FB} = \sum_{i=1}^n P_i^{FB} - (n-1)r_0 \quad (8)$$

where $R_i = r_0 - \alpha D_i$. P_i and R_i refer to the corresponding profit and revenue gained through each service provider i , respectively. Profit is simply generated from subtracting the compensations from the revenue. h_i is the outside option for provider i , which can be assumed to be zero without losing the generality. Substituting the critical value of $E[S_i]$ in Eq. (7), that is $h_i + k_i Var[S_i]/2 + V_i(A_i)$, into the objective function Eq. (6), the optimal time to be moved up is:

$$A_i^{FB} = \alpha / c_i \quad (9)$$

It can be concluded that the optimal time ahead of deadline depends positively on the marginal incentive from the customer, i.e. α , and is inversely proportional to cost coefficient c_i . Then the expected compensation to i -th provider $E[S_i^{FB}]$ and the revenue of enterprise P_{ep}^{FB} are:

$$E[S_i^{FB}] = \alpha^2 / 2c_i \quad (10)$$

$$P_{ep}^{FB} = \sum_{i=1}^n (r_0 - \alpha d_i + \alpha^2 / 2c_i) - (n-1)r_0 \quad (11)$$

These results indicate that when there is no private information between the principal and the agent, the optimal approach for the enterprise is to offer a fixed-price payment contract. This compensation is the sum of the outside options (which is zero in this case) and marginal cost of additional work of the agents. The enterprise bears all

the risks, since the risk premiums of the agents, i.e. $k_i \text{Var}[S_i]/2$ is zero, because $\beta_i^{FB} = 0$.

2.3. Second-Best Solution

The first-best case represents an ideal scenario. However, in reality, agents would not share all their private information. In this situation, “Incentive” implies that the offers the enterprise gives to the service providers should not only motivate them to participate in the project, but also trade off some of the risks that they would not like to bear. In this case, the agents can choose their optimal time ahead of deadline to maximize their net profits. The problem then becomes:

$$\max_{s_i} P_i^{SB} = E[R_i] - E[S_i] \quad (12)$$

$$\text{s.t. } E[S_i] - k_i \text{Var}[S_i]/2 - V_i(A_i) \geq h_i \quad (13)$$

$$A_i \in \arg \max_{A_i} E\{E[S_i] - k_i \text{Var}[S_i]/2 - V_i(A_i)\} \quad (14)$$

$$P_{etp}^{SB} = \sum_{i=1}^n P_i^{SB} - (n-1)r_0 \quad (15)$$

Compared with the first-best model, additional constraint Eq. (14) is added such that the agents can choose their own optimal time ahead of deadline subject to the cost of additional work and of risk aversion. Substituting the value of $E[S_i]$, $V_i(A_i)$ and $\text{Var}[S_i]$ into Eq. (14), taking first derivatives with respect to A_i , the optimal time ahead of deadline is:

$$A_i^{SB} = \beta / c_i \quad (16)$$

Then the expected payment to i -th provider $E[S_i^{SB}]$ and the revenue of enterprise P_{etp}^{SB} in this case are:

$$E[S_i^{SB}] = k_i \beta^2 \sigma_i^2 / 2 + \beta_i^2 / 2c_i \quad (17)$$

$$P_{etp}^{SB} = \sum_{i=1}^n \left(r_0 - \alpha d_i + (2\alpha\beta_i - \beta_i^2) / 2c_i - k_i \beta_i^2 \sigma_i^2 / 2 \right) - (n-1)r_0 \quad (18)$$

In this case, when cost information is private, the choice of optimal time ahead of deadline of the agent A_i^{SB} is proportional to the incentive intensity β_i . The compensations are different from the first-best case, which are the sum of the outside options, cost of additional work $c_i A_i^{SB} / 2$, and the risk premium $k_i \beta_i^2 \sigma_i^2 / 2$. In other words, the

enterprise trades off some of the risks that the agents are not willing to take with some incentives to induce them to exert additional work.

3. Analysis and Discussion

This section mainly analyzes and discusses how the agents respond to different compensation plans, and how the incentive structure is affected by different factors.

Remark 1 For enterprise pursuing a non-negative profit, the compensation function should not be steeper than the revenue function itself, i.e. $\beta_i \leq \alpha$.

Remark 1 identifies a simple design condition for enterprise that wants to make a non-negative profit. In other simple words, the incentive margin designed for each agent should not exceed the revenue margin received from the customer for the enterprise.

3.1. Information Symmetry and Information Asymmetry

As can be seen in Table 1, the optimal time ahead of deadline in the second-best case is less than those in the first-best case. However, the compensation to each service provider is increased, because of the existence of risk premium. It can be explained that: when there is no hidden information, the compensations are directly based (in the first-best case), the agent always chooses to conduct more additional work to acquire more payment, the principal bears all the risks. However, in the second-best case, the principal attempts to offer more incentives for more additional work to maximize its overall profit, since the profit stochastically increases with the additional works. The agents bear some of the risks they would not like to take. However, they are motivated by the increased payments.

Table 1. Comparison of Two Cases

	First-best Solution	Second-best Solution
A_i	α / c_i	β_i / c_i
$E[S_i]$	$c_i A_i^2 / 2$	$c_i A_i^2 / 2 + k_i \beta_i^2 \sigma_i^2 / 2$
P_i	$r_0 - \alpha d_i + \alpha A_i - c_i A_i^2 / 2$	$r_0 - \alpha d_i + \alpha A_i - c_i A_i^2 / 2 - k_i \beta_i^2 \sigma_i^2 / 2$

The value of information can be easily exhibited from Table 1, by subtracting P_i in the second-best case from the one in the first-best case, which is $(\alpha - \beta_i)^2 / 2c_i + k_i \beta_i^2 \sigma_i^2 / 2$. However in the real situation, we cannot guarantee all the organizations would like to share all their private information. As a result, the follows paper will analyze the information asymmetric case only.

3.2. Incentives in Outsourcing Contracts

Incentives in the outsourcing contract motivate the agents to exert additional work to finish their jobs ahead of the deadline. Since the revenue of the enterprise is also decreasing with the project duration, the more the agents speed up their working processes, the more revenue the enterprise will gain, subject to the cost of doing so. The principal will get extra profit of $\alpha A_i - c_i A_i^2 / 2 - k_i \beta_i^2 \sigma_i^2 / 2$ from each of the agent when including the incentives for early delivery of work in the contracts. The first part refers to the extra profit by early completion, while the last two terms are the costs, associated additional work and risk aversion respectively.

How to set the incentive intensity β_i becomes the most important problem. The logic can be refer to Figure 1, when increasing the incentive intensity β_i , each agent will put much more additional work, since the optimal value of A_i^{SB} is proportional to β_i . However when the value of β_i is increased, the compensation should also be increased, which should cover the cost of additional work and risk aversion. As a result, the optimal value of β_i should be the value that balance the increased profit and the compensation.

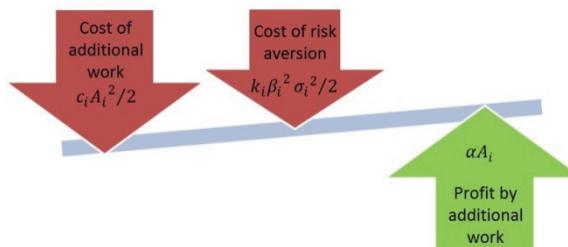


Figure 1. Profit of the Enterprise

Substituting A_i into P_i^{SB} , so that $P_i^{SB} = r_0 - \alpha d_i + (\alpha \beta_i - \beta_i^2) / 2c_i - k_i \beta_i^2 \sigma_i^2 / 2$. Taking the first derivative with β_i , so that the optimal value of incentive intensity is:

$$\beta_i^* = \frac{\alpha}{1 + c_i k_i \sigma_i^2} \quad (19)$$

It can be concluded that the optimal value of β_i is affected by many factors: the revenue margin of the customer α , the associated cost coefficient c_i and risk aversion coefficient k_i of different agents, as well as the external environmental component σ_i . It is more picturesque to use a simple numerical case to illustrate how the incentive intensity, choices of the agents, and profits of both the enterprise and agents interact with each other.

3.3. Numerical Example

To reveal the individual differences between different agents, it is assumed that the fixed payment to each agent s_{0i} , as well as the external environmental factor ε_i is the same. Three different values (small, medium, large) are assigned to two parameters: cost coefficient c_i and risk aversion coefficient k_i , respectively. So that there are 3^2 combinations. Nine agents with different cost structure and risk aversion are created.

It is assumed that the revenue function of the principal is in the form of $R_{ep} = 10000 - 200D_{tot}$, indicating a value of 200 bonus for every day ahead of deadline, 10000 is the base price. The incentive contracts to the service providers are: $S_i = 1000 - \beta_i(d_i - A_i + \varepsilon_i)$, where $d_i = 3$ referring that the scheduled task duration of each agent is three days. For simplicity, σ_i^2 equals 0.04 for all the agents. $c = (50, 100, 150)$ and $k = (0.2, 0.5, 0.8)$.

Figure 2 illustrates the differences between actual task durations and scheduled ones under different incentive intensities β . The difference increases linearly with β_i , and inversely related to the cost coefficients. Agent with smaller cost coefficient will finish the jobs much faster than those with larger cost coefficients.

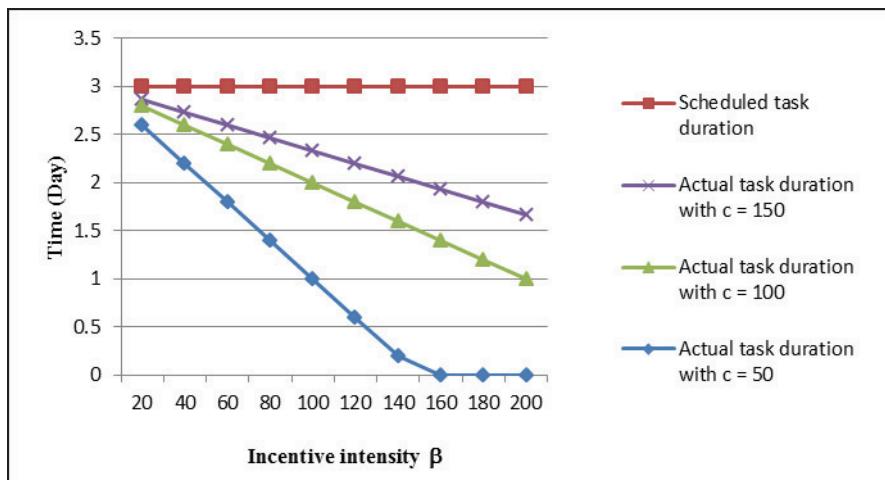


Figure 2. Actual Task Completion Time

Table 2 shows the extra profit from additional work on each agent under different incentive intensities. Number in bold indicates the optimal value of β_i . Since compensation covers both cost to additional work and risk aversion, which may exceeds the bonus on early completion, there is negative value in the final profits.

Agent 1, 2, and 3 has the same cost coefficient of 50, with risk aversion coefficient equals 0.2, 0.5 and 0.8 respectively. Repeatedly, $c = 100$ for agent 4, 5, and 6, and $c = 150$ for agent 7, 8, and 9. Comparing agents with the same cost coefficient, such as agent 1, 2, and 3, optimal value of incentive intensity is inversely proportional to risk aversion coefficient. For agents with same risk aversion coefficient, such as agent 1, 4,

and 7, optimal value of incentive intensity is inversely proportional to cost coefficient. These results validate the optimal incentive intensity β derived in Eq. (19).

The case with agent 7 with $c_7 = 150$ and $k_7 = 0.2$ is shown in Figure 3. Bonus for early completion linearly increases with the incentive intensity. However the total compensation increases exponentially with regard to incentive intensity. The optimal value of β/α making the most extra profit is 0.45. This result is consistent with the one calculating using Eq. (19).

Table 2. Extra Profit from Additional Work

	Percentage of β_i / α										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	0.0	74.4	137.6	189.6	230.0	260.0	278.4	285.6	272.6	245.4	215.0
2	0.0	72.0	128.0	168.0	192.0	200.0	192.0	168.0	119.0	51.0	-25.0
3	0.0	69.6	118.4	146.4	153.6	140.0	105.6	50.4	-34.6	-143.4	-265.0
4	0.0	36.4	65.6	87.6	102.4	110.0	110.4	103.6	89.6	68.4	40.0
5	0.0	34.0	56.0	66.0	64.0	50.0	24.0	-14.0	-64.0	-126.0	-200.0
6	0.0	31.6	46.4	44.4	25.6	-10.0	-62.4	-131.6	-217.6	-320.4	-440.0
7	0.0	23.7	41.6	53.6	59.7	60.0	54.4	42.9	25.6	2.4	-26.7
8	0.0	21.3	32.0	32.0	21.3	0.0	-32.0	-74.7	-128.0	-192.0	-266.7
9	0.0	18.9	22.4	10.4	-17.1	-60.0	-118.4	-192.3	-281.6	-386.4	-506.7

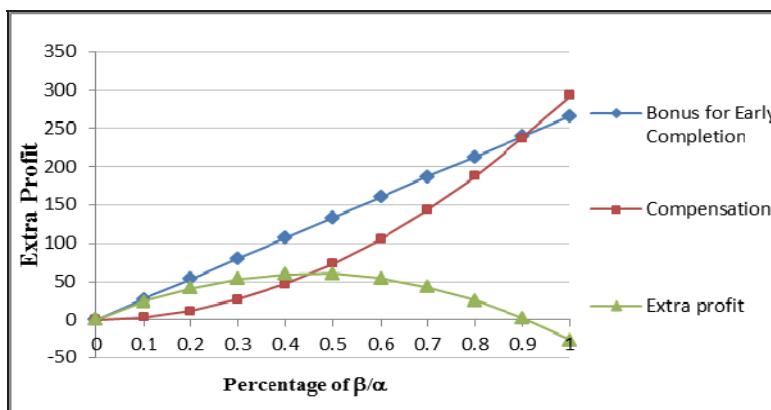


Figure 3. Extra Profit of agent

At last, the problem is solved as the principal offer the nine agents with different incentive intensities: $\beta^T = \{0.71, 0.50, 0.38, 0.56, 0.33, 0.24, 0.45, 0.25, 0.17\}$ in the each incentive contract. The whole project is originally supposed to be finished in 27 days. However the actual time is 17.2 days, which is 36.3% faster than the original time. The principal gets 981.9 units of bonus from the customer, account for 14.97% of the fixed price, which is 6560.

4. Conclusion

This paper uses agency theory to analytically investigate the optimal incentive contracts that an outsourcing enterprise should offer to their external logistics service providers. In this study, we contribute methodologically to the multi-task principal-agent literature by including a time-cost trade-off. Agents take different actions under different incentive mechanisms. Incentive contracts based on the agency theory help to motivate agents to take appropriate actions to realize the goal of the principal. These incentives serve the dual function of rewarding productive efforts and allocating risks. Different from traditional time-based incentive contracts, risk aversions of agents are also considered, making the time-cost trade-off more complex.

The findings highlight the importance for the principal to identify various factors that affect the behaviors of their agents, such as the marginal profit from the customer, external environmental uncertainties, and the heterogeneity of the agents. The heterogeneity is in terms of the responses toward risks and costs associated with speedup business process. Recommendations are made for the outsourcing enterprise to cooperate with external service providers through incentive contracts. Future research should look into a more general case: for example, the agents can be risk-aggressive, risk-averse, or prudent; the incentives can be linear, concave, convex, or in other more complicated forms; and the model may include the measurement of quality of service provided by the agents.

Acknowledgement

The work described in this paper was partial supported by the Government Research Fund (GRF) of the Research Grants Council of the Hong Kong Special Administrative Region, China (RGC # CityU 113609 and RGC # HKU 718809).

References

- [1] A. Marasco, Third-party logistics: A literature review, *International Journal of Production Economics* **113**(1)(2008), 127-147
- [2] M.S. Sohail, Sohal, A. S., The use of third party logistics services: a Malaysian perspective, *Technovation* **23**(5)(2003), 401-408
- [3] R. Bourles, D. Henriet, Risk-sharing Contracts with Asymmetric Information, *Geneva Risk and Insurance Review* **37**(1)(2012), 27-56
- [4] P. Dubois, T. Vukina, Optimal incentives under moral hazard and heterogeneous agents: Evidence from production contracts data, *International Journal of Industrial Organization* **27**(4)(2009), 489-500
- [5] I. Giannoccaro, P. Pontrandolfo, Supply chain coordination by revenue sharing contracts, *International Journal of Production Economics* **89**(2)(2004), 131-139
- [6] M. Raith, Specific knowledge and performance measurement, *Rand Journal of Economics* **39**(4)(2008), 1059-1079
- [7] B. Holmstrom, P. Milgrom, Multitask principal agent analysis - incentive contracts, asset ownership, and job design, *Journal of Law Economics & Organization* **7**(1991), 24-52
- [8] T.A. Taylor, E.L. Plambeck, Supply chain relationships and contracts: The impact of repeated interaction on capacity investment and procurement, *Management Science* **53**(10)(2007), 1577-1593
- [9] G.P. Cachon, M.A. Lariviere, Supply chain coordination with revenue-sharing contracts: Strengths and limitations, *Management Science* **51**(1)(2005), 30-44

A Multi-Agent System for Games Trading on B2B Market Based on Users' Skills and Preferences

Pavle SKOCIR^{a,1} and Gordan JEZIC^a

^a*Dept. of Telecommunications, Faculty of Electrical Engineering and Computing,
University of Zagreb, Croatia*

Abstract. Since the amount of content offered to users by service providers is rapidly rising every day, it is not an easy task for users to find content which would fit their interest. To deal with that problem, personalization services are being developed which monitor users' content consumption in order to try to offer content which is believed to correspond to preferences of a certain user. Alongside the problem that it is difficult for users to find the appropriate content, it is also difficult for service providers to procure and offer content for which users could be interested in. In this paper we focus on one content type - games for mobile phones and propose a model which enables analyzing data collected while users are playing a game. By using the results of the analysis, our model enables recommending new games to users, and also, what is the focus of this paper, acquiring new games which would fit users' skills and preferences. By purchasing distribution rights for new games which are believed to be appropriate for users, service provider can be more certain that the offered games would be widely adopted.

Keywords. multi-agent system, business-to-business market, personalization services, users' skills and preferences

1. Introduction

Due to large amount of different kinds of content that service providers offer to their users, it is difficult for those users to find games, videos, songs etc. which would fit their interests. On the other hand, it is also difficult for service providers to acquire and offer the appropriate content to its users. This problem can be dealt with by introducing personalized services which take into account users' individual needs, interests, preferences etc. to increase users' chances of finding the right information [1].

The mentioned problem also applies to content available to users of smart phones, since competition in the mobile market is shifting towards applications, services and usability instead of mere hardware [2]. In this transformation, the key players such as application developers, content providers and advertisers are putting more focus on analyzing user data in order to gain information about which type of content is preferred by users.

¹Corresponding Author: Pavle Skocir, University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia; E-mail: pavle.skocir@fer.hr

Games as a content type are no exception to the trend of introducing new services which monitor user behavior [3]. According to research conducted in UK and USA in 2011, games for smart phones are becoming very important for overall gaming market since even 44% of the people examined stated that they play game most often on their smart phones. Percentage of people who play games mostly on computers was 30% and on video console 21% [4]. As a result, companies which distributed games for personal computers or video consoles are starting to focus on smart phones [5].

This paper presents an upgrade of the system MARS, Multi-Agent Recommendation System for Games on Mobile Phones [6]. The MARS system analyses users' behavior in the game and recommends other games which are believed to be of interest to a specific user. In this paper we describe implemented and upgraded model for purchasing content distribution rights based on users' skills and preferences is described. Furthermore, recommendation model of the MARS is also upgraded by introducing new parameters into user analysis which will be described in detail in Section 3.

For content trading implementation in B2B markets, software agents which automate business interactions and act on behalf of their owners are suitable to be used [7]. Various models can be implemented for negotiation between agents, such as multi-attribute auctions where agents negotiate over different parameters [8]. To decide which offer is the most suitable one, e.g. user preferences or content provider reputation can be taken into account [9].

In our research, the process of new games acquisition is initiated according to the preferences and skills of the users by which we try to ensure that the new games would suit them. The decision about the game to be purchased depends on the revenue percentage set by content provider.

The paper is organized as follows. Section 2 describes the model of the proposed system for game provisioning. Section 3 provides an insight into the system architecture. Section 4 presents the business model for ordering new games on B2B market as well as calculation of the parameters used for modeling users' skills and preferences. Section 5 concludes the paper.

2. The Model of the Game Provisioning System

The game provisioning system proposed in this paper enables recommending games for mobile phone users and acquiring new games which best fit users' skills and preferences. The model of the system shown in Figure 1 is based upon four main processes: playing games, analyzing game consumption data, recommending games and purchasing game distribution rights. While a user plays a game, the system monitors her/his activities on her/his mobile phone which concern the game downloaded from the service provider. Collected information is sent to service provider and analyzed afterwards. The results of the analysis are used to recommend new games to users as well as to enhance the decision making process for purchasing content distribution rights for new games from various content providers. Once the content provider delivers the games to the service provider, new recommendation lists are generated and games are recommended to users in correspondence with users' skills and preferences. If a user likes the recommended game, she/he downloads it and plays it on her/his mobile phone.

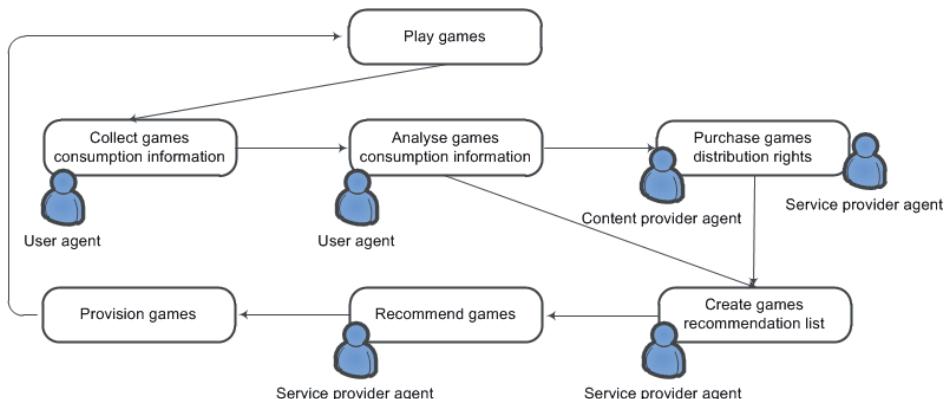


Figure 1. The Model of the Recommendation and Game Acquisition System

In our model we used software agents because of their ability to automate business interactions and negotiate on behalf of their users [7]. As seen in Figure 2, there are three types of agents in the system: Service Provider Agent (SPA), Content Provider Agent (CPA) and User Agent (UA). Their behaviour and tasks will be explained in detail in the following section.

3. Architecture and Implementation of the Multi-Agent System for Game Provisioning

The proposed model is implemented using agent platform JADE² (*Java Agent DEvelopment Framework*). Agents in the Multi-Agent System for Game Provisioning enable trading on two different electronic markets (e-markets): business-to-customer (B2C) and business-to-business (B2B) content e-market. In this paper we have processed trading on B2B market, while B2C trading by recommending games to users is described in our previous work [6].

In our system, three types of agents are defined: Service Provider Agent (SPA), Content Provider Agent (CPA) and User Agent (UA). The SPA represents a service provider on the content e-market which offers games and other content types to users. The CPAs represent game publishing companies or individuals which publish and produce games and act as content providers on the B2B content e-market. On the B2C content e-market, the SPA provides game-based services to users which are represented by their UAs. The system enables automated trading on the B2B and B2C content e-markets. Based on users' skills and preferences analysis, the SPA recommends suitable games to its users and, based on the gameplay statistics, it purchases content distribution rights from CPAs for new similar games.

Digital games purchased by user from the service provider are located on her/his mobile phones, as well as the UA. The UA monitors user's interaction with the game. It keeps track of the time user spends playing the game and certain levels of it, as well as the number of times that user started the game. It sends collected information to SPA which

²<http://jade.tilab.com/>

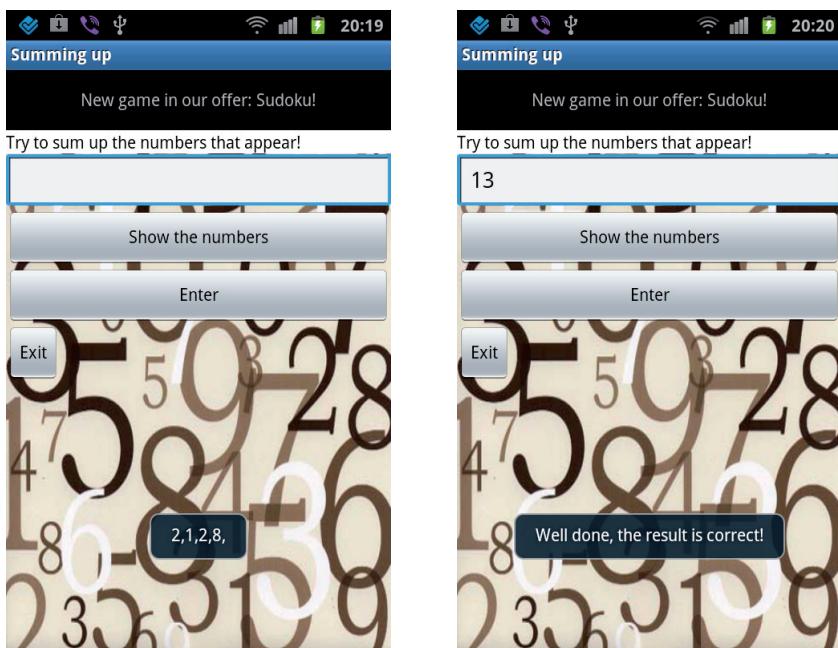


Figure 2. The implemented addition game

analyzes it. We introduced following parameters for monitoring users' skills and preferences: progress, motivation and success. SPA determines the values of before mentioned parameters based on the information received from UA.

Upon the analysis of information received from UA and determination of progress, motivation and success levels, SPA carries out recommendations based on analyzed users' skills and preferences. It also takes into account information specified in the user profile and the profile of the users' mobile phone. The SPA sends new games recommendations periodically to UA which displays game information to the user. Additionally, the SPA calculates motivation, progress and success levels for different genres of the games offered by service provider which will be used for new content acquisition. Periodically, the SPA contacts the CPA and negotiates the purchase of content distribution rights for new games. Methods for calculating progress, motivation and success levels for a user playing a certain game, as well for different genres of the games will be presented in the following section.

To test our proposed system, we implemented a simple game for Android OS in which users have to add the set of numbers shown on the screen, as shown in Figure 3. In each higher level there is one more number in the set. After adding shown numbers, user writes the result into a box.

4. Business Model for Ordering New Games on B2B Market

Proposed model enables service provider to obtain new games on B2B market. Service provider tries to acquire games which are believed to be of high interest to users. Knowl-

edge about types of games for which users are believed to be interested in is gained by analyzing data collected for each user during times of playing different games.

User experience, i.e. how user feels during playing a certain game, is determined by assessment of following parameters: progress, motivation and success. Values of each of the mentioned parameters are used later as input parameters for the process of ordering new games.

In this chapter, developed methods for assessing progress, motivation and success levels are explained. Later on, process for ordering new games on B2B market using progress, motivation and success level is presented.

4.1. Progress

Progress for each user i in the game j is determined by monitoring times required for crossing a level k through a number of attempts to play the game:

$$t_{ijk} = t_{ijks} + \sum_{f=0}^p t_{ijkf} \quad (1)$$

where t_{ijks} represents the playtime of the level k of the game j when user i successfully completed it, while t_{ijkf} represents the playtime of the level k when the user i failed to successfully complete it assuming that p is the total number of failed attempts.

Progress p_{ij} is calculated using the following expression:

$$p_{ij} = \sum_{k=1}^m w_{ijk} p_{ijk}, \text{ with condition } \sum_{k=1}^m w_{ijk} = 1 \quad (2)$$

where p_{ijk} presents progress level for each level k of the game j and w_{ijk} is weight factor for each level which is calculated according to the number of times user played a certain level. Parameter p_{ijk} is calculated according to time t_{ijk} and can take three values which explain whether the user is making progress (times t_{ijk} to complete a certain level k are improving), whether there is no more room for improvement (a certain level k cannot be passed in less time t_{ijk}) or whether user is not making progress (times t_{ijk} to complete a level k are oscillating). It is assumed that if user is making progress, the game is still challenging and interesting enough. If there is no more room for improvement, it is assumed that the game is mastered and that user could be willing to play a new, potentially more challenging game. By monitoring progress, we monitor flow, i.e. we want to estimate if the game is in the observed moment challenging enough for the user and that it suits her/his skills.

4.2. Motivation

Motivation of the user in the game is determined by collecting information about the number of times the user played the game and total time spent during playing. Motivation is calculated using the expression:

$$m_{ij} = w_n m_{ij}^n + w_r m_{ij}^r + w_t m_{ij}^t, \text{ with condition } w_n + w_r + w_t = 1 \quad (3)$$

where m_{ij}^n represents motivation level based on the number of attempts to play the game, m_{ij}^r is motivation level based on the multiple playing of certain game level without exiting the game, while m_{ij}^t represents motivation level based on the total time spent while playing the game.

Motivation level m_{ij}^n is calculated using the collected information about each starting of the game in the observed period. Since research shows [4] that a mobile phone game is averagely played three to four times a month, we assumed that if user plays a game four times a month, the motivation level is average. If a certain game is played more, this motivation level is assumed to be higher and vice versa.

To calculate motivation level m_{ij}^r , we took into account information about how many times the user played certain levels in the game without exiting. We assumed, as with motivation level m_{ij}^n , that the user is averagely motivated if she/he had done so three to four times a month.

To calculate motivation level m_{ij}^t , average times that user spent playing the game \bar{t}_{ij} is compared to average time that all users spent playing the game \bar{t}_j . Time \bar{t}_{ij} is defined as:

$$\bar{t}_{ij} = \frac{1}{n} \sum_{l=1}^n t_{ij}^l = \frac{1}{n} \sum_{l=1}^n \sum_{g=1}^r t_{ij}^{lg} \quad (4)$$

where t_{ij}^l is time during l^{th} from all together n times the game is played. During l^{th} playing, user could have played the game r times from the beginning until the end. Duration of each such playing is described with parameter t_{ij}^{lg} . Time \bar{t}_j is defined using the expression:

$$\bar{t}_j = \frac{1}{u} \sum_{i=1}^u \bar{t}_{ij} \quad (5)$$

where i represents each from u users who plays game j . If time \bar{t}_{ij} is similar to time \bar{t}_j , it is assumed that user i is averagely motivated. If time \bar{t}_{ij} is higher than time \bar{t}_j , it is assumed that the user i is highly motivated and vice versa.

By analyzing before mentioned parameters we try to determine if the user is highly, moderately or poorly motivated for playing the observed game. By adjusting weights w_n , w_r and w_t we gave the highest impact to overall motivation for user i to motivation level m_{ij}^n , followed by m_{ij}^r and m_{ij}^t . It is assumed that if the user is highly motivated, than that particular game and similar ones are interesting to her/him and she/he would likely play them. On the contrary, if the motivation is poor, it is assumed that the user would prefer playing a game of other thematic.

4.3. Success

Success of the user playing a game is determined by collecting information about the number of successfully and unsuccessfully finished levels of the game as well as comparing times needed to pass a certain level with times of other users. It is calculated using the expression:

$$s_{ij} = \sum_{k=1}^m w_{ijk} s_{ijk}, \text{ with condition } \sum_{k=1}^m w_{ijk} = 1 \quad (6)$$

where k is a level of the game, m is the total number of levels. s_{ijk} is success of user i in level k which is calculated using the expression:

$$s_{ijk} = w_r s_{ijk}^r + w_t s_{ijk}^t, \text{ with condition } w_r + w_t = 1 \quad (7)$$

where s_{ijk}^r is parameter of the success level which is defined in relation to the ratio of times that user passed a certain level and total number of times that user played that level. Success level s_{ijk}^t is defined in relation to average time \bar{t}_{ijk} needed for user to pass a level k of the game j and average time \bar{t}_{jk} needed for other users to pass a level k of the game j . Time \bar{t}_{ijk} is defined as follows:

$$\bar{t}_{ijk} = \frac{1}{n} \sum_{l=1}^n t_{ijk}^l = \frac{1}{n} \sum_{l=1}^n (t_{ijk_s}^l + \sum_{f=1}^p t_{ijk_f}^l) \quad (8)$$

where n is a total number of attempts to play the game which resulted with passing the level. Each of the l attempts is described with time $t_{ijk_s}^l$ needed for user i to pass the level k of the game j and p times $t_{ijk_f}^l$ which describe times of playing the game level which did not result in mastering it. Time \bar{t}_{jk} is described similarly as \bar{t}_{ijk} , only that average time is calculated for other users h in the system from the total number of users u :

$$\bar{t}_{jk} = \frac{1}{u} \sum_{h=1, h \neq i}^u \frac{1}{n} \sum_{l=1}^n (t_{ijk_s}^l + \sum_{f=1}^p t_{ijk_f}^l) \quad (9)$$

Mentioned times \bar{t}_{ijk} and \bar{t}_{jk} are compared. If the average time \bar{t}_{ijk} is similar to the average time \bar{t}_{jk} , it is assumed that user i is equally successful as other players. If the time \bar{t}_{ijk} is better than time \bar{t}_{jk} , it is assumed that the user is more successful than others and if the time \bar{t}_{ijk} is worse than time \bar{t}_{jk} , it is assumed that the user is less successful than others.

Weights from Eq. (7) are adjusted so that both success levels s_{ijk}^r and s_{ijk}^t have the same significance. By calculating value s_{ij} we try to determine if the observed user i is less, more or averagely successful comparing to all users playing the game j . It is assumed that if the user is less successful, she/he is not satisfied with the currently played game and a less demanding game would be more appropriate. On the other hand, if the user is more successful than the others, a more demanding game would be appropriate.

4.4. Process of ordering games on B2B market

Each game in the proposed model is attached to a certain category (e.g. action games, puzzles etc.) and difficulty (easy, medium and difficult). We try to determine category and difficulty of the games which are most popular, i.e. for which the motivation level is the highest. Motivation level $m_{g,d}$ for each category c and difficulty d is determined using the following expression:

$$m_{g,d} = \frac{1}{u+v} \sum_j^v \sum_i^u m_{ij} \quad (10)$$

where v is the number of games of the observed category and u is total number of players who are playing certain game j . Parameter m_{ij} represents a motivation level for game j played by user i defined using Eq. (3). When motivation levels for each category and difficulty are determined, progress and successfulness level $ps_{g,d}$ for observed category and difficulty is calculated using the expression:

$$ps_{g,d} = \frac{1}{u+v} \sum_j^v \sum_i^u \frac{p_{ij} + s_{ij}}{2} \quad (11)$$

where p_{ij} represents progress level and s_{ij} successfulness level for user i and game j . Parameter $ps_{g,d}$ is calculated as mean value of progress level p_{ij} and successfulness level s_{ij} described in Eq. (2) and (6) respectively, and represents the level of difficulty d of the certain game to the users and is used to define the difficulty $d_{searched}$ of the game which will be tried to be obtained from content providers. Procedure of the assessment of the searched difficulty is shown in Algorithm 1.

Algorithm 1 Defining difficulties of the game to be searched

```

switch psg,d do
  case (psg,d ≤ 1.5)
    | if (d ≤ 2) then dsearched = 1 else dsearched = 2
  case (psg,d > 1.5) && (psg,d < 2.5)
    | dsearched = d
  case psg,d ≥ 2.5
    | if (d < 3) then dsearched = d + 1 else dsearched = d
  
```

Parameter $ps_{g,d}$ is assigned to values from interval $ps_{g,d} \in [1, 3]$. Difficulty of the game d in the proposed system is assigned to values $d = \{1, 2, 3\}$. If value $ps_{g,d} < 1.5$, it is assumed that users are not making progress nor are they successful in playing games of category g and difficulty d , i.e. times required to pass a certain level are not improving and they need a lot of attempts to successfully pass that level. Therefore it is concluded that observed difficulty is too high and that it would be appropriate to order less demanding games. If value of the parameter $ps_{g,d}$ is between $ps_{g,d} \in [1.5, 2.5]$, it is assumed that the users are successful and making progress, i.e. times required to pass a certain level are improving and not many attempts are needed to do so. Therefore it is concluded that the observed difficulty is appropriate for those users and that it would be appropriate to procure games of the same difficulty because they fit users' skills. If the value of the parameter $ps_{g,d}$ is between $ps_{g,d} \in [2.5, 3]$, it is believed that users do not have any room for improvement, they are extremely successful because they need little time to pass a certain game level which cannot be reduced and they need little number of attempts to pass the game level. Therefore it is concluded that it would be more appropriate for those users to play games of higher difficulty.

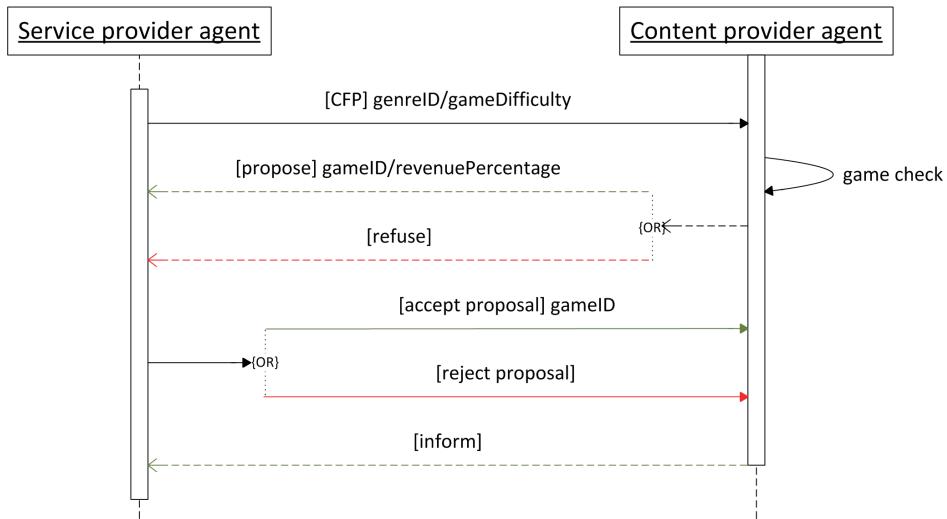


Figure 3. Process of new games acquisition on B2B market

When motivation levels $m_{g,d}$ are determined for each category and difficulty of the game to be searched $d_{searched}$, the process of provisioning new games from content provider can be started. Service provider publishes a demand for games of category with highest motivation level $m_{g,d}$ and of difficulty $d_{searched}$. Content providers offer games of desired category and difficulty and demanded a certain revenue percentage. By using such negotiation model, service provider ensures to have in its offer the games for which high interest is shown by users. At the same time, content providers can negotiate higher revenue percentage than they could get by publishing games independently in the application store.

Negotiation is implemented by using software agents and Contract Net protocol specified by FIPA³ organization (Foundation for Intelligent Physical Agents). Communication between Service Provider Agent (SPA) and Content Provider Agents (CPA) is shown in Figure 3.

In the implemented system there is one SPA and more CPAs. Communication using Contract Net Protocol consists of four parts. Firstly SPA sends request for games of desired category and difficulty (CFP, *Call For Proposal* message). Secondly, each CPA checks if it has specified game ready for distribution. If it does, it sends information about the offered game alongside with desired revenue percentage (message *propose gameID/revenuePercentage*). On the other side, if content provider does not have the specified game, it declines proposal by sending message *refuse*. After receiving all proposals, SPA finds the best proposal, which is the one in which content provider offers the highest percentage revenue for service provider which will be collected after offering the game to users. SPA accepts the best offer by sending message *accept proposal*, while others are rejected by sending message *reject proposal*. Finally, CPA whose proposal is accepted sends the game to SPA under arranged conditions.

³<http://www.fipa.org/>

5. Conclusion and Future Work

We presented a Multi-Agent System for trading with games for mobile phones which enables new games acquisition based on skills and preferences of the users. The system collects data about users' playing of each game and analyses users' skills and preferences by measuring three parameters: progress, motivation and success. We described how those parameters are calculated and used to procure new games. By doing that, the problem of being difficult to find games which fit users' skills and preferences is diminished because such games are acquired from content providers and then recommended to users in order to make the game discovery easier.

For future work, new games should be developed and evaluation of the system should be performed by letting various users play games, and the system to recommend new ones. Recommendations can be regarded as successful if the users download and play recommended games. As the next step, the evaluation of the process of acquiring new games which is described in detail in this paper should be elaborated.

To purchase game distribution rights from content providers, skills and preferences of all users are calculated. For future work it would be worth to enable grouping users according to their skills and preferences and purchasing different games for each group.

Acknowledgments

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

References

- [1] Lee, D.H.; Brusilovsky, P.: Fighting Information Overflow with Personalized Comprehensive Information Access: A Proactive Job Recommender. In: Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems, pp. 21-26. IEEE, Athens (2007)
- [2] Verkasalo, H.: Analysis of Smartphone User Behaviour. In: Proceedings of the 9th Internetional Conference on Mobile Business, pp.258-265. IEEE, Athens (2010)
- [3] Weber, B.G.; Mateas, M.; Jhala, A.: Using Data Mining to Model Player Experience, Workshop on Evaluating Player Experience in Games, ACM, Bordeaux (2011)
- [4] Information Solutions Group, PopCap games mobile phone gaming research, 2012, http://www.infosolutionsgroup.com/2011_PopCap_Mobile.Phone_Games.Presentation.pdf
- [5] Feijoo, C.; Gomez-Barroso, J.-L.; Aguado, J.-M.; Ramos, S.: Mobile Gaming: Industry challenges and policy implications. Telecommunications Policy, 36, 3, pp. 212-221 (2012)
- [6] Skocir, P., Marusic, L., Marusic, M., Petric, A.: The MARS A Multi-Agent Recommendation System for Games on Mobile Phones. In: KES-AMSTA 2012. LNAI, vol. 7327, pp. 104-113. Springer, Heidelberg (2012)
- [7] Podobnik, V.; Petric, A.; Jezic, G.: An Agent-based Solution for Dynamic Supply Chain Management. Journal of Universal Computer Science. 14, 7, pp. 1080-1104 (2008)
- [8] Petric, A.; Jezic, G.: Multi-attribute Auction Model for Agent-Based Content Trading in Telecom Markets. In: KES 2010, Part I. LNCS, vol. 6276, pp. 261-270. Springer, Heidelberg (2010)
- [9] Petric, A.; Jezic, G.:A Multi-Agent System for Game Trading on B2B Electronic Market. In: KES-AMSTA 2011. LNCS, vol. 6682, pp. 332-341. Springer, Heidelberg (2011)

Intelligent Agents with Semantic Technology (IAST 2013)

Automatic Reuse of Interaction Protocols in Mas: Arip Model

Sami KHALFAOUI^{a,1} and Wided LEJOUAD CHAARI^a

^a*Soie, National School of Computer Studies, University of Manouba, Tunisia*

Abstract. The traditional agent architecture is essentially based on three major components, namely, its competences, its roles, and its interactions. In a previous work, we proposed to dissociate the interactions from the agent content, and then to consider them as separated and shared resources among agents. In this way, multi-agent systems are called Interaction-Centered MAS where interactions are easily updated, maintained and reused independently without changing the agent source code. In this context, the system requires an Automatic Reuse of Interaction Protocols (ARIP). In this paper, we describe the ARIP model, its new concepts (like protocol description schema, agent description schema, etc.), and its main functionalities making possible the reasoning on the interaction protocols and their automatic reuse.

Keywords: Multi-agent system, interaction protocol, interaction pattern, automatic reuse, interaction server.

1. Introduction

Interaction is a key concept in multi-agent systems. It expresses the dynamic of the agents. The good management of interactions is a major issue of MAS. In previous work [1] and [2], we have been interested in the expression of interactions and we proposed the interaction based model which separates the interactions from the internal architecture of the agents. In this approach, we consider the interactions as first class entities that can be stored in a warehouse and reused in different contexts. This leads us to propose a complete mechanism for the Automatic Reuse of Interaction Protocols: ARIP.

Many works are interested in the reuse of the protocols of interaction in the multi-agent systems [3], [4] and [5]. We noticed that these works focused on the expression of the interactions to facilitate the reuse. They introduced formalisms and languages to describe the interactions without proposing mechanisms for the reuse. Other works like [5] tried to propose an approach for the reuse of interaction protocols in which the protocol selection is done at design time.

¹ Corresponding Author: Sami Khalfaoui, Soie, National School of Computer Studies, University of Manouba, Manouba, Tunisia; E-mail: Sami.khalfaoui@topnet.tn

In the approach presented in [6], an interaction situation is described using several criteria. These criteria will be used later by the multi-agent system designer to select the suitable protocol for the interaction situation.

To date, only a few research works tackled the issue of dynamic selection of interaction protocols. [7] presented a method which enables agents to select protocols by themselves at runtime when they need to interact with one another. In the present work, we start by constructing a matrix of protocols that can be suitable for the collaborative task. Then the initiator agent tries these protocols until it reaches its goal. In this approach, an error recovery process is presented to allow agents to select another protocol from the matrix.

In our approach, we present a more detailed description of both protocol and agent for better and more precise protocol selection.

In this paper, we begin with a brief reminder of the interaction-centered approach. Then we introduce new concepts which will serve as a basis for the automatic selection of an interaction protocol by an agent. We describe at the end the complete mechanism for the automatic reuse of interaction protocols.

2. Interaction-Centered Approach

The interaction-centred approach separates the interactions from the code of the agents. This allows separating the intrinsic behavior of the agents from the communication between them. The interactions are expressed on a higher level of abstraction as interaction patterns. These interaction patterns are described in the declarative language ISL (Interaction Specification Language) [8]. The ISL is independent of the application language. It includes several operators such as the conditional operator (if . . . then . . . else . . . endif), the sequential operator (:), the concurrency operator (//), waiting operators and an exception handling operator. An interaction pattern defines at least one interaction rule. Interaction rules express the control that should be executed on the connected components. An interaction rule consists of two parts: the left side is the notifying message and the right side is the action. The semantic of an interaction rule is to rewrite method code. That is, instead of executing the default method (default behaviour), the interaction runtime should execute the actions specified in the rule's action.

Figure 1 describes a simple example of an interaction pattern defined between two agent classes “SellerAgent” and “BuyerAgent”.

```

1 interaction sell_buy (SellerAgent seller, BuyerAgent buyer)
2 {
3     buyer.buy(book) -->
4         Boolean available := seller.checkAvailability (book);
5         if(available) then
6             seller.sel(book) // buyer._call
7 }
```

Figure 1. An example of interaction pattern.

Interaction patterns are defined between agent classes and stored in a warehouse. Then, the interaction patterns are instantiated between agents needing to interact (see Figure 2). For this purpose, we defined a mechanism for dynamic instantiation of interaction patterns between the agents.

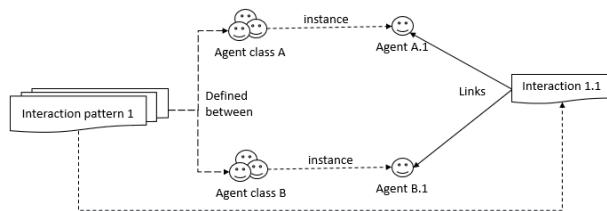


Figure 2. From interaction patterns to interactions.

The interaction-centred approach is an interesting basis for the reuse of interaction protocols. We enriched our model with new concepts making possible the reasoning on the protocols of interaction and their automatic reuse.

3. Interaction Protocol Description

A crucial step to reuse interaction protocols is to be able to identify them. Indeed, the agent, depending on the problem to resolve, must identify the right protocol.

To identify a protocol, we associate it with the following informations:

- A name: it is the protocol identifier. It must be unique and significant (for example, English bid).
- A parent protocol: identifies the parent protocol in the case of an inheritance.
- A context: indicates the domain of the protocol (for example, e-commerce, medicine, etc.).
- A goal: describes the motivation of the protocol (for example: buy a book, make a reservation, etc.). The definition of a goal will be described in the section 3.2.
- A set of roles: describes the roles that the agents can play within the protocol (for example, for the protocol "English bid ", the roles that the agents can play are: the bidder and the participant). The description of a role will be presented in the next section.
- An initiator: it is the role initiator of the protocol.
- A behavior: it is the control structure of the protocol. It consists of an interaction pattern expressed in the ISL language.

The description of an interaction protocol is called Protocol Description Schema (PDS).

To express the PDSs we opted for the XML language [9]. Indeed, this language is understandable as well by the designers as by the agents. Furthermore, XML allows

the automatic generation of documentation thanks to XSLT. It is also independent from platforms and allows a high degree of interoperability between various applications.

Figure 3 shows an example of PDS related to the protocol for the booking of a hotel room.

```

1 <?xml version='1.0' encoding='UTF-8'?>
2 <ProtocolDescriptionSchema>
3   <protocol_name>hotel_reservation</protocol_name>
4   <parent/>
5   <context>accommodation</context>
6   <objective>book room</objective>
7   <roles>
8     <role>client</role>
9     <role>booking_service</role>
10   </roles>
11   <initiator>client</initiator>
12   <behavior>hotel_reservation.isl</behavior>
13 </ProtocolDescriptionSchema>
```

Figure 3. Example of a PDS.

In this example, the attributes “objective” and “roles” were briefly presented. The detail of these attributes will be exposed later in this section.

In the “behavior” attribute we indicated that the behavior of the protocol is described in the file "hotel_reservation.isl". This assumes that the ".isl" file is created and stored in the warehouse of the interaction patterns.

An interaction protocol is thus constituted by a behavior expressed in ISL which will be stored in the interaction patterns warehouse, and a description (PDS) which will be stored in an interaction protocols directory. To implement this directory, we also opted for the XML language.

3.1. Role Description

As mentioned in the previous section, one component of a PDS is the set of roles which the agents can play within the protocol. This concept has been discussed in many works [10], [11] and [12].

In our context, a role is described by the couple <**Role_name**, **Operations**> where:

- **Role_name** is the name of the role,
- **Operations** is the list of the operations that an agent, playing this role, must perform. These operations can be seen as conditions to play a role by an agent within a protocol.

Roles are important criteria in the selection process of a protocol by an agent. The mechanism of protocol selection will be detailed in the section 4.

3.2. Goal Expression

In a multi-agent system, every agent has a goal. To reach this goal, the agent must select the right protocol. Indeed, the protocol is also characterized by a goal which describes the problem resolved by this protocol.

To automatically select an interaction protocol, an agent must be able to perform a matching between its own goal and those described in the interactions protocols. To do this, goals must be expressed in a homogeneous way. Indeed, several works in the field of requirements engineering [13], [14], [15], [16] and [17] have studied the goals

expression. According to [14] and [17], a goal consists of a verb, a target and one or more parameters. Figure 4 shows the goal structure adopted in our approach.

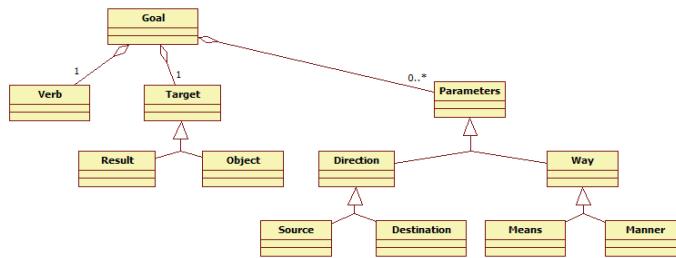


Figure 4. Goal structure.

As shown in Figure 4, the formulation of a goal includes a verb, a target and parameters said “direction” or “way” parameters. The first two elements are mandatory while other parameters are optional.

- The **target** refers to the entity impacted by the goal. The formulation of a minimum goal has a verb and the target parameter. For example, “Book a room” where “room” is the target.

We distinguish two types of target: the **object** and the **result**. An object exists before the realization of the goal while a result ensues from the satisfaction of the goal. So in “Make a reservation”, the reservation is the result of the realization of the goal, while in “Confirm a reservation”, a reservation is an object because it exists before the satisfaction of the goal.

- The **direction** parameters characterize the **source** and the **destination** of the entities connected to the goal. For example:
 $\text{Make}_{\text{verb}} (\text{a reservation})_{\text{object}} (\text{from a request})_{\text{source}}$
 $\text{Store}_{\text{verb}} (\text{a reservation})_{\text{object}} (\text{in the database})_{\text{destination}}$
- The **way** parameters refer to the instruments and approaches for achieving a goal. The **means** describes the instrument to achieve the goal. For example:
 $\text{Pay}_{\text{verb}} (\text{reservation})_{\text{object}} (\text{by credit card})_{\text{means}}$
The **manner** identifies how the goal can be reached. For example:
 $\text{Cancel}_{\text{verb}} (\text{a reservation})_{\text{result}} (\text{by expiration of the waiting period})_{\text{manner}}$

3.3. Behavior Expression

The behavior concerning an interaction protocol corresponds to an interaction pattern expressed in ISL. This interaction pattern is defined between agent classes. However, in a PDS, the participants to a protocol are not described in terms of classes of agents but in terms of roles so, interaction patterns have also to consider the roles.

Indeed, the interaction protocols are not anymore described between agent classes but between roles as shown in Figure 5.

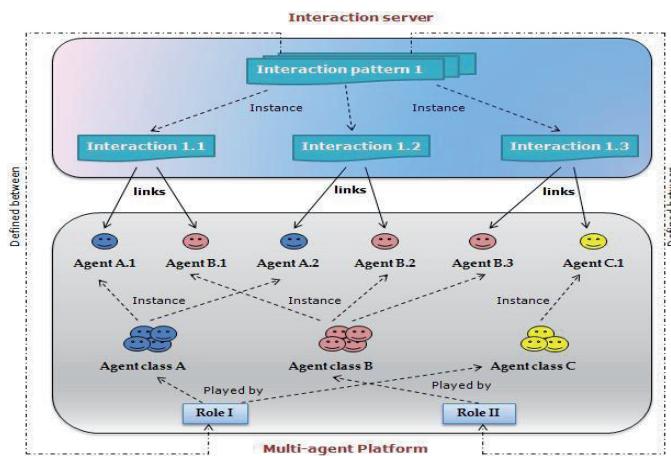


Figure 5. Definition of interaction patterns between roles.

In the example above, the interaction pattern 1 is defined between the Role I and the Role 2. Let's consider the interaction 1.1. This interaction is instantiated between the agents A.1 and B.1. The agent A.1 is an instance of the agent class A which plays the role I and the agent B.1 is an instance of the agent class B which plays the role II.

To express interaction patterns defined between the roles we shall use the same syntax ISL presented in the section 2. This is possible because the roles and the agents classes associated to these roles have the same signatures of methods.

The mechanism of interaction pattern instantiation will not be detailed in this paper.

3.4. Agent Structure

We suggest associating to an agent a description facilitating the research and the selection of an interaction protocol. We call this description Agent Description Schema (ADS).

ADS := <Agent_class, Context, Objective, Operations>

- **Agent_Class** is the class of the agent described by this ADS,
- **Context** is the domain for which the agent was designed. The context can refer to an ontology of domain in which are described all the concepts concerning a domain. The “context” will be taken into account during the selection process of a protocol.
- **Objective** is the goal of the agent. This goal will be described according to the structure presented in the section 3.2.
- **Operations** is the list of the operations that the agent can perform. The attribute “operations” presents the signatures of the public methods of the agent.

In the same way as for the operations related to roles (section 3.1), every operation of the list “Operations” is described by <Method_name, parameters, Return_type>.

The operations will be useful to the agent to verify if it is able to play a specific role within a protocol.

Besides an ADS, we integrated into the agent an engine allowing it:

- To select a protocol from the directory of the interaction protocols,
- To instantiate the selected protocol,
- To identify the agents participating in the interaction(s),
- To verify if they are able to participate to the interaction(s).

4. The Arip Mechanism

Previously, we presented the different concepts necessary for the automatic reuse of interaction protocols. In this section, we expose the mechanism for automatic selection of a protocol by an agent. Figure 6 shows the ARIP model.

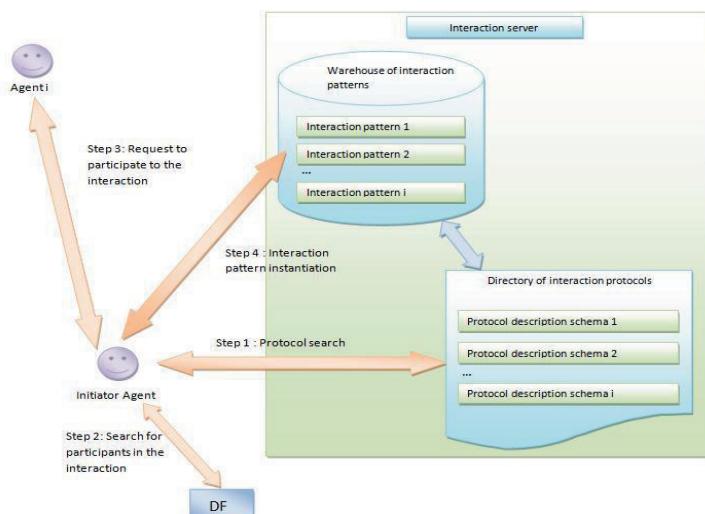


Figure 6. The ARIP model.

For the correspondence between contexts and objectives we use a semantic matching. This correspondence is facilitated by the descriptive layers added as well to the agent as to the protocol (ADS and PDS). The semantic matching has been discussed in multiple works especially in the field of Web Services [18], [19] and [20]. We were inspired by these works to propose a simple approach based on a thesaurus [21] and [22] in which every term is associated with a lexical field

Step 1: Protocol search.

The agent begins by checking if its context corresponds to the context of the protocol. If this is the case, the agent goes to the mapping of its goal with the goal of the protocol. This correspondence is primarily structural, i.e. that the goals should have the same attributes. An objective with the attributes (verb, object, way) cannot correspond to an objective with the attributes (verb, result, means). Once the objectives match structurally, the agent uses the thesaurus to verify the semantic correspondence between every attribute. So, the goal “buy book by credit card” is equivalent to the objective “get book by bank card”.

The third filter in the search for a protocol is based on the ability of an agent to play the initiator role in the selected protocol. This is a matching between the signatures of agent methods (described in the ADS) and the functionalities described in the initiator role description.

At the end of this stage, the agent selects the first protocol that meets the three filters mentioned above.

Step 2: Search for participants in the interaction.

In this stage, the agent initiator will try to find participating agents in the interaction. To do it, it sends its request to the Directory Facilitator DF. The DF [23] is an agent who acts as yellow pages within standardized FIPA multi-agents platforms. The DF contains the list of services offered by every agent. This allows answering the requests of the agents which look for a particular service.

In our context, we enriched the structure of the DF by adding ADSs structured in directory. So, an agent sends to the DF a request containing the description of a role for which it wants to associate an agent. The DF checks in the role descriptions and the ADSs, to provide to the requesting agent a list of agents that can play the requested role.

Step 3: Request to participate to the interaction.

Once the initiator has received the list of agents that can participate in the interaction, it sends them requests to participate to the interaction. The concerned agents can accept or decline the request.

In the case of refusal, the initiator agent consults the next agent in the list supplied by the DF. If the initiator cannot find participants to the protocol, the research process returns to step 1.

Step 4: Interaction pattern instantiation.

In this step the initiator agent instantiates the interaction between itself and other agents.

Figure 7 presents a summary of the steps presented above.

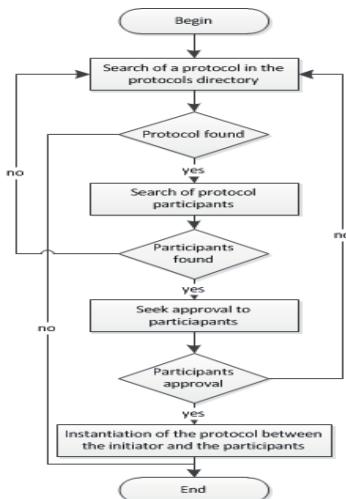


Figure 7. Selection process of an interaction protocol.

5. Conclusion

In this paper, we proposed an approach for the automatic reuse of interaction protocols called ARIP. In this approach, an interaction protocol is composed by a description called PDS stored in a directory of protocols and a behavior expressed in ISL and stored in a warehouse. The directory of protocols and the warehouse of interaction patterns compose the interaction server. This approach required also the addition of a descriptive layer to the agents called ADS. The ADS and the PDS help the agent to select the suitable protocol to achieve its goal. In this first version of ARIP, we supposed that the goals of the agents are atomic, and to every goal corresponds one protocol. However, it is frequent that the goal of an agent is decomposed into sub-goals and for each of these sub-goals corresponds a particular protocol. The next version of ARIP will propose a means to compose automatically interaction protocols to reach the global goal of the agent.

References

- [1] S. Khalfaoui, W. Lejouad Chaari and A-M. Pinna Dery, Interactions entre composants pour environnements multi-agents, Journée Multi-Agents et Composants, Paris 2005.
- [2] S. Khalfaoui, and W. Lejouad Chaari, Réification des interactions entre agents, 15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Tours, 2006.
- [3] C. Hanachi, and C. Sibertin-Blanc, Protocol Moderators as Active Middle-Agents in Multi-Agent Systems, Autonomous Agents and Multi-agent Systems, Vol. 8. (2004), 131-164.
- [4] T. Doi, Y. Tahara, and S. Honiden, IOM/T: an interaction description language for multi-agent systems, In 4rd International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, 2005.
- [5] P. Mathieu and S. Picault, Vers une représentation des comportements centrée interactions, 15e Rencontres Francophones Reconnaissance des Formes et Intelligence Artificielle, 2006.

- [6] S. Bussmann, N. R., Jennings, and M. Wooldridge, Re-Use of Interaction Protocols for Agent-Based Control Applications, In the 3rd international conference on Agent-oriented software engineering, Bologna, 2002.
- [7] J. Quenum, S. Aknine, O. Shehory, and S. Honiden, Dynamic Protocol Selection in Open and Heterogeneous Systems, Intelligent Agent Technology, Hong Kong, 2006.
- [8] M. Blay, D. Ensellem, A. Charfi, A-M, Pinna, M. Riveill. Software Interactions. , in Journal of Object Technology, vol. 3, no. 10, (2004), 161–180.
- [9] D. Hunter, D. Ayers, J. Rafter, E. V. Der Vlist, and J. Fawcett, Beginning XML, Wrox Pr Inc,2007.
- [10] P. Mathieu, RIO: Roles, Interactions and Organizations, 3rd International Central and Eastern European Conference on Multi-Agent Systems, Prague, 2003.
- [11] G. Cabri, L. Ferrari, and F. Zambonelli, Role-Based Approaches for Engineering Interactions in Large-Scale Multi-agent Systems, Lecture Notes in Computer science, (2004), 243-263.
- [12] J. Odell, H. Marian and L. Renato, A Metamodel for Agents, Roles, and Groups, Agent-Oriented Software Engineering V, 5th International Workshop, New York, 2004.
- [13] A. Anton, Goal-based requirements analysis, 2nd International Conference on Requirements Engineering, Colorado, 1996.
- [14] N. Prat, Goal formalisation and classification for requirements engineering, 3rd International Workshop on Requirements Engineering: Foundations of Software Quality, Barcelona, 1997.
- [15] E. Yu, Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, 3rd IEEE, 1997.
- [16] C. Rolland, N. Prakash and A. Benjamen, A. A multi-Model View of Process Modelling, Requir. Eng. Vol. 4, (1997), 169-187.
- [17] N. Prat, Réutilisation de la trace par apprentissage dans un environnement pour l'ingénierie des processus, Phd. Université Paris1, Paris, 1999.
- [18] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, Semantic matching of webservices capabilities. First International Semantic Web Conference, 2002.
- [19] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, S-Match: an algorithm and an implementation of semantic matching, In Proceedings of ESWS'04, 2004.
- [20] A. Doan and A. Halevy, Semantic integration research in the database community: A brief survey, AI Magazine, Special Issue on Semantic Integration, 2005.
- [21] J. Morris and G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text, Computational Linguistics, Vol.17 (1), (1991), 21–48.
- [22] M. Okumura and T. Honda, Word Sense Disambiguation and Text Segmentation Based on Lexical cohesion, 5th International Conference on Computational Linguistics, Tokyo, 1994.
- [23] FIPA, FIPA Agent Management Specification, 2002, Retrieved June 2, 2012, from <http://www.fipa.org/specs/fipa00023/SC00023J.html>

A Method for Knowledge Integration Using Indeterminate Model of Time with Criterion O₂

Van Du NGUYEN^{a1} and Ngoc Thanh NGUYEN^b

^a*Nong Lam University, Ho Chi Minh City, Vietnam.*

^b*Institute of Informatics, Wroclaw University of Technology, Poland.*

Abstract. In this paper we will investigate processing inconsistency of knowledge and its integration process using the temporal model of indeterminate valid time and probability. This paper differs from others by considering criterion O₂ in designing algorithm for integration of knowledge. In the integration process, we need to determine a proper time interval and probability (*based on consensus method*) which properly represent an event. For this aim, some postulates, algorithm for knowledge integration are worked out and analyzed.

Keywords. consensus method, indeterminate valid time, knowledge integration

1. Introduction

For processing knowledge in distributed environments, integration is a very important process since knowledge arising in autonomous sources often contains inconsistency, incompleteness and uncertainty. Owing to integration of knowledge from different sources one can get more complete information about some objects. Processing inconsistency of knowledge (or conflict processing) is the most important and difficult task in knowledge integration. Up to now, we have two criteria for processing inconsistency of knowledge based on consensus method. They are: “*1-Optimality*” and “*2-Optimality*” or O₁ and O₂ for short. O₁ means the sum of distances between the consensus and the profile elements to be minimal. While O₂ means the sum of the squared distances between a consensus and the profile elements is minimal. Although criterion O₁ is more popular than O₂ for consensus choice and some optimal tasks [1], criterion O₂ is better than O₁ in some situations.

Model of indeterminate valid time serves to describe the timestamp and possibility of events that take place in the future. Data model supporting indeterminate valid time has been investigated by Dyleson and Snodgrass in [2,3]. In [4-6] the authors have applied this model for processing inconsistency of knowledge based on consensus method. The knowledge structure multi-attribute and multi-valued has been investigated in those works. In [7] attribute probability has been considered in model indeterminate valid time. In this approach, however, attribute probability is considered

¹ Corresponding Author: Van Du Nguyen, Nong Lam University, Ho Chi Minh City, Vietnam. Email: nvdu@hcmuaf.edu.vn.

as certainty degrees of an agent in giving statement about occurrence of an event and the result of consensus method contains time interval.

In this work, we investigate another aspect of above model by considering probability as certainty degrees about occurrence of an event in the future. In addition, criterion O_2 is considered in the integration process. Applying model indeterminate valid time to process conflicts in many practical situations, we would like to obtain a time interval and probability that best represent occurrence of an event. For example, *three meteorological stations place in neighbour areas give statements about landfall time of typhoon KHANUN to Korea as follows: station in Taiwan, agent states that landfall time is [19/07/2012-23/07/2012] and its probability is 90%; while according to station in Korea, landfall time is [16/07/2012-17/07/2012] with probability 80% and station in Japan, landfall time is [18/07/2012-21/07/2012] with probability 60%. The question is: "What interval and probability are correct?"* That is the main problem what we will investigate in this work.

The rest of this paper is organized as follows. Section 2 presents related works that used data model indeterminate valid time. In section 3, data model support indeterminate valid time and probability is presented. In section 4, problem of integration, postulates for knowledge integration are considered. Integration algorithm is proposed in section 5. Finally, some conclusions and future works are presented in section 6.

2. Related works

Model indeterminate valid time serves to describe the timestamp and possibility of events that take place in the future. Its applications in processing inconsistency of knowledge that arisen in the integration process based on consensus method have been investigated in many works [4-7]. These works can be distinguished into two groups depend on using criterion in designing integration algorithms. They can be called: O_1 or O_2 . The first of them consists of "*1-Optimality*" [4-7]. In the second group, "*2-Optimality*" criterion was considered [7]. Therefore, in this section we present an overview of those works in which model indeterminate valid time was applied.

For O_1 criterion, it means the sum of distances between the consensus and the profile elements to be minimal. This criterion is very popular not only in consensus theory but also in optimization theory [1]. In [5], the author has applied this criterion for processing inconsistency of knowledge with data model indeterminate valid time. Distance function of kind: minimizing transformation costs for measuring difference between time intervals has been proposed. In this approach, however, the role of each agent as well as possibility about occurrence of events was equivalent. It also means probability is 100%. Subsequently, in [4,6], a deeper investigation in this data model has been worked out. The authors proposed three distance functions of kind: reflecting element shares in the distance and some algorithms for determining a proper representation of a conflict profile. These functions were designed by considering the participation of each chronon in relationship between time intervals. In [7], the author considered appearance of attribute probability (*data model <From, To, Prob>*). This is necessary because in many practical situations, occurrence of events in the future is not always 100%. It should have smaller probability. However, attribute probability is considered as certainty degrees of agents in giving statements about occurrence of an event in the reality. It means that, each agent has been assigned a weight value and all

the statements of an agent have the same probability value. Besides, the results of algorithms contain only time interval.

Similarly for O_2 , the sum of the squared distances between a consensus and the profile elements is minimal. This criterion was investigated in [6]. In that work, the author has applied this criterion for processing inconsistency of knowledge for data model indeterminate valid time. However, attribute probability was not considered. The aim of algorithm is: determine a proper time interval that best represents a conflict profile.

In general, the common problems of above mentioned works are: attribute probability did not appear or appeared but was not taken into account in designing distance functions between complex tuples. Almost above mentioned works used criterion O_1 for designing integration algorithms. In addition, the results of algorithms only return time intervals.

In this work, we will investigate another aspect of this data model by assuming appearance of probability attribute as certainty degrees about occurrence of an event. In addition, determining a proper time interval and probability that best represent the conflict profile is considered by criterion O_2 . This approach is missing in the literature.

3. Data model

In this section, we will consider some notions about time, time interval and probability as well as modal relation in processing inconsistency of knowledge.

3.1. Time structure

Applying model of time in processing inconsistency of knowledge has been investigated by Nguyen in [4-7]. This model is adopted from the model of Knight and Ma [8,9], Dyereson [10], Allen [11] and Gadia [12]. In this section we present some basic concepts in these works: the time-line segment is partitioned into a finite number of smaller segments; each called a chronons [13-15]. A chronon is the smallest amount of time that can be represented in the implementation. A time interval i is a contiguous set of chronons, should be identified by two parameters: lower bound chronons i_* and upper bound chronons i^* ($i^* \geq i_*$), and is denoted by $i = \langle i_*, i^* \rangle$. The length $d(i)$ or $|i|$ of interval i is measured by the number of chronons lying between i_* and i^* is equal to $(i^* - i_* + 1)$. A time point is special interval i where $(i^* = i_*)$. In addition, it is possible to use numerical operations: '+' or '−', relations as: '>', '=', or '<' on chronons labels and operations '∩' and '÷' between intervals. NOW² is an abstract chronon representing the current instant of time [5], can be identified by the system.

The timestamp of occurrence of some events are an instant on the time line. An instant is presented by the label of the chronons in which the instant is located. However, to represent an event that takes place in the future we do not know the exact chronons during which an instant is located, but we know between which chronons an instant is located. We call that instant is an indeterminate instant. The chronons between which an instant is located are called lower and upper supports; thus they create an interval.

² In this work, the value of NOW we accept day 16 July 2012.

3.2. Probability measure

Probabilistic approaches have been used by many authors in database system [16-19]. In this paper, we assume that probability is a percent value and belongs to $[0, 1]$. In addition, we can perform mathematical operation as ‘+’ and ‘-’ and relations as ‘>’, ‘=’, or ‘<’ between two values. In this paper, however, when an agent or expert states about occurrence of an event with probability pr we assume that it is the certainty degrees that event will take place in the given time interval.

3.3. Kinds of events

In [5], the author has classified three kinds of events: *historical, open and supposed event*. In this work, we consider a new kind of events based on supposed event that is, “*point event*”. As its name suggests: an event is called “*point*” means that event will only occur in a point of given time interval and its occurrence is only once. This kind of events is very popular in the real world such as: *falling of rocket/satellite or landfall of typhoon etc.* Therefore, when an agent gives opinion about occurrence of an event in a time interval with probability pr , we have two assumptions about occurrence of the event at the outside of that time interval as follows: agent does not know whether event occurs or not; event occurs with probability $100\% - pr$. The second assumption is a slightly complicated situation in integration process because of determining which intervals conflict together. Thus, in this work we do not consider this situation. In addition, we use modal relation to describe events that take place in the future. Modal relation is one of the temporal relational schemas. Data model $\{Event, Scenario, From, To, Prob\}$ is called a modal schema if all the values v of temporal attributes *From*, *To* fulfill the condition $v > NOW$.

4. Integration problem and postulates for knowledge integration

The integration problem is defined as follows: given a conflict profile:

$$X = \{r_j = (i_j, pr_j) : i_j \in I \wedge pr_j \in [0, 1], j = 1, 2, \dots, n\} \quad (1)$$

It is necessary determine a tuple $r^* = (i^*, pr^*)$ of type $T \in A$ which best represents the given tuples. (Where $T = \{From, To, Prob\}$, $A = \{Event, Scenario, From, To, Prob\}$).

Some notions related to integration process such as: I , I_{pr} , $\Pi(I_{pr})$, $\hat{\Pi}_k(I_{pr})$ have the same meaning in what we mentioned in the master thesis [20].

In this section we propose some postulates for knowledge integration based on consensus method. By $Con(I_{pr})$ we denote the set of all consensus functions for time intervals and probabilities I_{pr} .

Definition 1. By a consensus function for profiles we understand a function

$$C : \hat{\Pi}(I_{pr}) \rightarrow \Pi(I_{pr}) \quad (2)$$

which satisfies one or more of the following postulates.

Let $X, X1, X2 \in \hat{\Pi}(I_{pr})$, $\partial_{pr}(y, X) = \sum_{y \in I_{pr}} \partial_{pr}(z, y)$ for $z \in I_{pr}$ we have:

P1. Reliability (*Re*) iff

$$\left(\forall X \in \hat{\Pi}(I_{pr}) \right) (C(X) \neq \emptyset)$$

P2. Unanimity (*Un*) iff

$$(C(n^*x) = x)$$

P3. Condorcet consistency (*Cc*) iff

$$\left(\forall X1, X2 \in \hat{\Pi}(I_{pr}) \right) [(C(X_1) \cap C(X_2) \neq \emptyset) \Rightarrow (C(X_1 \cup X_2) = (C(X_1) \cap C(X_2)))]$$

P4. Quasi-unanimity (*Qu*) iff

$$(x \notin C(X)) \Rightarrow (\exists n \in \aleph : x \in C(X \cup (n^*x)))$$

P5. Proportion (*Pr*) iff

$$\left(\forall X1, X2 \in \hat{\Pi}(I_{pr}) \right) ((X_1 \subseteq X_2) \wedge (x \in C(X_1)) \wedge (y \in C(X_2))) \Rightarrow (\partial_{pr}(x, X_1) \leq \partial_{pr}(y, X_2))$$

P6. 1-Optimality (*O₁*) iff for any

$$(x \in C(X)) \Rightarrow \left(\partial_{pr}(x, X) = \min_{y \in U} \partial_{pr}(y, X) \right)$$

P7. 2-Optimality (*O₂*) iff for any

$$(x \in C(X)) \Rightarrow \left(\partial_{pr}^2(x, X) = \min_{y \in U} \partial_{pr}^2(y, X) \right)$$

P8. Closure (*Cl*) iff

$$i^* \subseteq \bigcup_{(i, pr) \in X} i$$

P9. Fairness (*Fa*) iff

$$(X \text{ is homogeneous}) \Rightarrow C(X) = \{(i, pr^*)\} : pr^* = \frac{\sum_{(i, pr) \in X} pr}{\text{card}(X)}$$

X homogeneous means all time intervals are identical.

P10. Probability Simplification (*Ps*) iff

$$(\forall (i, pr), (i', pr') \in X : pr = pr' = pr^*) \Rightarrow (C(X) = \{(i^*, pr^*) : i^* \in C'(X)\})$$

for some $C' \in \text{Con}(I)$

An element of set $C(X)$ is called a consensus of profile X .

Some commentaries for defined postulates: postulate P1 (Re) states that a consensus must exist for each non-empty profile. Postulate P2 (Un) requires that if a profile is homogeneous then the only consensus is the element belonging to this profile. It is a very intuitive requirement for many consensus choice tasks. Postulate P3 (Cc), the common part of two consensuses of two sets should be a subset of the consensus of the sum of these sets. It means that if some intervals are simultaneously the consensus of two interval sets, then it should also be the consensus of the sum of these sets. According to postulate P4 (Qu), if an element x is not a consensus of a profile X , then it should be a consensus of a new profile X' containing X and n elements x for some n . Postulate P5 (Pr), as its name Proportion suggests, is a natural condition because if the profile is bigger, then the greater difference is between its consensus and its elements. Postulates P6 (O_1) is a standard condition for consensus choice and should be useful for integration determining. This criterion is very natural and popular in consensus determining. Its justification is based on the requirement that the integration should best represent the given opinions of the agents. Thus it should minimally differ from these opinions. Postulate P7 (O_2) states that the sum of the squared distances between a consensus and the profile elements to be minimal. The main idea of postulate P8 (Cl) is the common part of opinions which should be included in the component of consensus. The purport of this postulate is similar to Pareto criterion: If all voters vote for the same candidate then he should be finally chosen. According to postulate P9 (Fa), if profile X is homogeneous then the consensus of X will receive that time interval and average of those probabilities. Postulate P10 (Ps) is also an intuitive situation. When agents give statement about occurrence of an event with the same probability but in different time intervals, then the consensus of profile will be the consensus of profile and that probability value. In the next section, we consider relationship among postulates. In this paper, however, we mainly analyze relationship between postulates O_2 with other postulates.

Theorem 1. Let's consider the following statements about relationship among postulates:

1. The following statement is true:

$$O_2 \cap Re \subseteq Co \cap Qu \cap Un$$

2. The following statements are not true:

- (a) $O_1 \cap O_2 \cap Re \neq \emptyset$

- (b) $Re \cap Co \cap Qu \cap Pr \cap O_1 \cap O_2 \neq \emptyset$

Proof.

1. The first statement is true according to [1,6]. This is very important properties of consensus choice functions specifying postulates O_2 . If a consensus function satisfies postulate O_2 , it can satisfy many other postulates.

2. To prove this statement let X be a profile: $\{(<1, 2>, 0.8), (<7, 8>, 0.6), (<9, 10>, 0.9)\}$ and distance function we have the following results: $O_1: (<7, 8>, 0.6)$ while $O_2: (<6, 6>, 0.6)$. Meaning statement a) is not true. The second statement is a result of statement (b).

5. Algorithm for knowledge integration

In order to propose algorithms for processing inconsistency of knowledge, the most important issues are distance functions. In the context of this paper we use three distance functions of kind “reflecting element shares in the distance” to measure difference between complex tuples that mentioned in the master thesis [20]. Concretely, we have:

The first function was proposed by considering the participation of each chronon that lies between time intervals is 100%.

$$\rho_{pr1}((i_1, pr_1), (i_2, pr_2)) = \begin{cases} |i_1| \times pr_1 + |i_2| \times pr_2 + \left(\frac{\max(i_{1*}, i_{2*})}{-\min(i_{1*}, i_{2*})} + 1 \right), & i_1 \cap i_2 = \emptyset \\ |i_1| \times pr_1 + |i_2| \times pr_2 - |i_1 \cap i_2| \times (pr_1 + pr_2) + \\ |i_1 \cap i_2| \times |pr_1 - pr_2|, & i_1 \cap i_2 \neq \emptyset \end{cases} \quad (3)$$

Similarly, the second function was proposed by assuming the participation of each chronon lies between time intervals is 200%.

$$\rho_{pr2}((i_1, pr_1), (i_2, pr_2)) = \begin{cases} |i_1| \times pr_1 + |i_2| \times pr_2 + \left(\frac{\max(i_{1*}, i_{2*})}{-\min(i_{1*}, i_{2*})} + 2 \right), & i_1 \cap i_2 = \emptyset \\ |i_1| \times pr_1 + |i_2| \times pr_2 - |i_1 \cap i_2| \times (pr_1 + pr_2) + \\ |i_1 \cap i_2| \times |pr_1 - pr_2|, & i_1 \cap i_2 \neq \emptyset \end{cases} \quad (4)$$

The last one, the author did not take into account chronons lie between time intervals.

$$\rho_{pr3}((i_1, pr_1), (i_2, pr_2)) = |i_1| * pr_1 + |i_2| * pr_2 - |i_1 \cap i_2| * (pr_1 + pr_2) + \\ |i_1 \cap i_2| * |pr_1 - pr_2| \quad (5)$$

In the next section we present an algorithm for determining a reliable consensus for a conflict profile. This algorithm is considered on satisfying postulate O₂, meaning that the value of algorithm is the minimum value of the sum of the squared distances between a consensus and the profile elements.

Algorithm 1. The idea of algorithm is based on every possible interval is included in the minimal interval that contains all of intervals from profile \tilde{X} and probability belongs to $[pr_{min}, pr_{max}]$ to determine a tuple (i, pr) that satisfies postulate O₂. The procedure for this algorithm is presented as follows:

Input:

- o X – conflict profile
- o $\delta \in \{\rho_{pr1}, \rho_{pr2}, \rho_{pr3}\}$ – distance function

Output:

- $C_{I_{pr}}$ – Consensus for conflict profile X, meaning that satisfies postulate O₂.

Procedure:

BEGIN

Step 1: Set Sum = MAX_VALUE.

Step 2: determine the smallest lower chronon and the greatest upper chronon from the set of time intervals (J) of profile X.

$$i_{min} = \min\{i_j : i_j = (i_{j*}, i_j^*) \in J\}, \quad i_{max} = \max\{i_j^* : i_j = (i_{j*}, i_j^*) \in J\}$$

Step 3: determine the smallest as well as greatest probability value from the set of probabilities (Pr) of profile X.

$$pr_{min} = \min\{pr_j : pr_j \in Pr\}, \quad pr_{max} = \max\{pr_j : pr_j \in Pr\}$$

Step 4: For $i_* = i_{min}$ to i_{max} doFor $i^* = i_*$ to i_{max} doFor $pr = pr_{min}$ to pr_{max} do

Begin

o Create interval $I_{pr} = ((l_*, l^*), pr)$ o Evaluate the sum of distance from I_{pr} to other intervals of X.

$$S = \partial(I_{pr}, X)$$

o If $S \leq Sum$, then $Sum = S, i_{pr} = l_{pr}$

End

End

Theorem 2.a) The complexity of algorithm 1 is $O(n^2 * m)$ where n is the length of interval, m is the length of pr_{min} to pr_{max} .b) The algorithm returns an optimal consensus satisfying postulate O₂.**Proof.** The proof of this theorem is given in the master thesis [20].**Example 1.** Let's consider conflict profile which mentioned in section 1. We have:**Table 1.** A conflict profile.

<i>profile(r₁)⁺</i>		
Station	Time interval <From, To>	Prob
Korea	16/07/2012 - 17/07/2012	0.8
Japan	18/07/2012 - 21/07/2012	0.6
Taiwan	19/07/2012 - 23/07/2012	0.9

Apply algorithms 1 for $profile(r_1)^+$ we have the result as follows:**Table 2.** the result with O₂.

Function	Time interval <From, To>	Prob
----------	--------------------------	------

ρ_{pr1}	19/07/2012-20/07/2012	0.6
ρ_{pr2}	18/07/2012-21/07/2012	0.6
ρ_{pr3}	19/07/2012-21/07/2012	0.6

In table 2, three functions of kind “reflecting element shares in the distance” are used to measure difference between complex tuples. We see that the results of algorithm are slightly different. This difference is the result of applying different distance functions. Therefore, users should choose a suitable distance function according to their situations. Meaning depend on conflict profile and relationship among time intervals users choose a proper distance function for processing inconsistency of knowledge.

6. Conclusions

This paper has presented another aspect of model indeterminate valid time and probability. The model in this work differs from others [4-7] in considering appearance of attribute probability as the certainty degrees about occurrence of an event and criterion O₂ has been considered in designing algorithm for knowledge integration. In addition, some relationship of other postulates with O₂ has been considered. The future works should be: proposing methods to choose a proper criterion (O₁ or O₂) for integration process; optimizing integration algorithms.

References

- [1] N. T. Nguyen, *Advanced Methods for Inconsistent Knowledge Management*. Springer, London, 2008.
- [2] C. E. Dyreson, and R. T. Snodgrass, Temporal Indeterminacy. In *The TSQL2 Temporal Query Language*, edited by R.T. Snodgrass, 475-499. Kluwer Academic Publish Hingham, 1995.
- [3] C. E. Dyreson, and R. T. Snodgrass, Supporting Valid-time Indeterminacy. *ACM Transaction on Database Systems* 23, no. 1 (1998), 1-57.
- [4] N. T. Nguyen, Consensus-based Timestamps in Distributed Temporal Databases. *The Computer Journal* 44, no.5 (2001), 398-409.
- [5] N. T. Nguyen, Representation Choice Methods as the Tool for Solving Uncertainty in Distributed Temporal Database Systems with Indeterminate Valid Time. *Lecture Notes in Artificial Intelligence*, 445-454. Springer-Verlag, 2001.
- [6] N. T. Nguyen, *Methods for Consensus Choice and Their Applications in Conflict Resolving in Distributed Systems*. Wroclaw University of Technology Press, 2002.
- [7] N. T. Nguyen, Modal Time and Processing Its Inconsistency in Temporal Data Collections. In *Ontologies and Soft Methods in Knowledge Management*, edited by R.Katarzyniak, 101-118. Adelaide, Australia: Advanced Knowledge International, 2005.
- [8] B. Knight, and M. Jixin, An Extended Temporal System Based on Points and Intervals. *Information Systems* 18, no. 2 (1993), 111-120.
- [9] B. Knight, and J. Ma, A General Temporal Theory. *The Computer Journal* 37 (1994), 114-123.
- [10] C. E. Dyreson, M. Soo, and R. T. Snodgrass, The Data Model for Time. In *The TSQL2 Temporal Query Language*, edited by R.T.Snodgrass, 327-346. Kluwer Academic Publish Hingham, 1995.
- [11] J. F. Allen, Maintaining Knowledge about Temporal Intervals. *Communication of the ACM* 26, no. 11 (1983), 832-843.
- [12] S. K. Gadia, A Homogeneous Relational Model and Query Languages for Temporal Databases. *ACM Transaction on Database Systems* 13, no. 4 (1988), 418-448.
- [13] G. Ariav, A Temporally Oriented Data Model. *ACM Transaction on Database Systems* 11, no. 4 (1986), 499-527.

- [14] J. Clifford, and A. Rao, A Simple, General Structure for Temporal Domains. In *Proceedings of the Conference on Temporal Aspects in Information Systems*, edited by C.Rolland, F.Bodart, and M.Leonard, 23-30. Association Francaise pour la Cybernetique Economique et Technique, Montreuil, France, 1987.
- [15] C. E. Dyreson, F. Grandi, W. Kafer, and N. Kline, A Consensus Glossary of Temporal Database Concepts. *ACM SIGMOD - SIGMOD Record* 23, no. 1 (1994), 52-64.
- [16] D. Barbara, H. Garcia-Molina, and D. Porter, A Probabilistic Relational Data Model. In *Proceedings of the 2nd International Conference on Extending Database Technology: Advances in Database Technology*, edited by F.Bancilhon, C.Thanos, and D.Tischritzis, 60-74. Venice, Italy, New York: Springer-Verlag, 1990.
- [17] D. Dey, and S. Sarkar, A Probabilistic Relational Model and Algebra. *ACM Transaction on Database Systems* 21, no. 3 (1996), 339-369.
- [18] N. Fuhr, and T. Roelleke, A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Transactions on Information Systems* 15, no. 1 (1997), 32-66.
- [19] E. Gelenbe, and G. Hebrail, A Probability Model of Uncertainty in Databases. In *Proceedings of the Second International Conference on Data Engineering*, 328-333. IEEE Computer Society Press CA, 1986.
- [20] V. D. Nguyen, *Methods for Processing Inconsistency of Knowledge in Multi-attribute and Multi-valued Structures*. M.Sc. Thesis (Advisor: N. T. Nguyen). Wroclaw University of Technology, 2012.

A Layered Adjustable Autonomy Approach for Dynamic Autonomy Distribution

Salama A. MOSTAFA^{a,1}, Mohd Sharifuddin AHMAD^b, Muthukkaruppan ANNAMALAI^c, Azhana AHMAD^b and Ghusoon Salim BASHEER^a

^a*College of Graduate Studies, Universiti Tenaga Nasional, Malaysia*

^b*College of Information Technology, Universiti Tenaga Nasional, Malaysia*

^c*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia*

Abstract. Autonomy adjustment to a system requires a mechanism to implement the roles of the autonomous entities in the system. The required degree of autonomy to which the autonomous entities adhere is a highly debated topic. On one hand, people argue that strict minimal autonomy to the autonomous entities is sufficient in producing reliable systems. On the other hand, others deliberate that the entities with full autonomic capabilities are essential aspects of advanced intelligent and flexible systems. The adjustable autonomic agent approach can be a solution for both cases. In this paper, we extend the idea of modeling a spectrum of autonomy in a layered structure, where the agents can act at different layers of autonomy in order to fulfill the system's autonomic conditions. Consequently, a logical representation of the conceptual model of Layered Adjustable Autonomy (LAA) is proposed. The LAA model aims to give the system implicit control over the agents' decisions whenever necessary by managing the agents' autonomy, ensuring quality and robust decision-making. An Autonomy Analysis (AAM) and Situation Awareness (SAM) Modules are proposed to attest the dynamic distribution of agent autonomic performance to a degree of autonomy level.

Keywords. Software agent, Adjustable autonomy, Layered Adjustable Autonomy (LAA), Dynamic autonomy distribution, Autonomy analysis, Situation awareness

1. Introduction

The main idea behind any intelligent autonomous system is to minimize human intervention in the system's operation [1]. To many realistic domains, the applicability of such system is constrained by many limitations [2]. However, multi-agent systems (MAS) paradigm reinforces the concept of autonomy to be much more significant [1].

Nevertheless, the practical setting of the MAS requires the behavior of the agents in the system subjected to each agent's local desire, as well as the system's global desire, which mandates dynamic distribution to the agents' autonomy [3]. This issue indicates the fact that agent autonomy is adjustable since interactions with others are required to manifest emerged intelligence. Adjustable autonomy that allows both interventional and non-interventional actions is desirable for many cases [2].

¹ Corresponding Author: Salama A. Mostafa, College of Graduate Studies, Universiti Tenaga Nasional, Putrajaya Campus 43000, Selangor Darul Ehsan, Malaysia; E-mail: semnah@yahoo.com.

Inyama et al. [4] state two views to adjustable autonomy: the client and the contractor (bilateral) or the degree of autonomy is either increased or decreased (bidirectional). The main reason of an agent autonomy level concurrently updated (internally or externally) is to meet different situations [5]. Falcone and Castelfranchi [6] claim that an explicit mechanism is needed to demonstrate different kinds of autonomy levels. This mechanism must be able to adjust agent autonomy level and provides dynamic distribution of control over the system [7].

In our previous work, we propose a flexible and usable Layered Adjustable Autonomy (LAA) model [7]. The aim of the proposed model is to provide dynamic autonomy distribution mechanism and works in a MAS environment. LAA enables an agent to perform at different levels of autonomy for different circumstances through layers of autonomy. Each layer selection condition is obtained based on the selected action type and agent autonomy conditions, where an action or a set of actions represents task achievement. In this work, we illustrate the logical representation of the LAA model in which agent's decision is distributed between full and non-autonomous dimensions. The autonomy distribution equips the agents to act at the appropriate layer of autonomy and directing the agent's performance as required.

2. Perceptions of Autonomy

The abstract definition of autonomy is the ability of making unaided decisions. Consequently, autonomy adjustment is a process of changing the parameters of agent decision-making, based on task performance, so as to influence the decision outcome. Many researchers have stated that an autonomous agent is intelligent, and so assume that the agent itself is capable of handling its autonomous behavior [8].

That means, when an agent responds to a particular event, it would autonomously pursue the corresponding tasks without oversight or intervention of control from others [9]. An issue with this assumption is that in some cases the agents are incapable, uncertain, unpredictable, unreliable and/or unauthorized for making decision when performing a given task [5], [11]. One way out is to recognize agents with varying degree of autonomy, i.e., agents that are fully-autonomous, semi-autonomous and non-autonomous [9], [10], [11].

Huber [8] emphasizes that agent's autonomy is imposed by external influences which makes it difficult to be defined only in terms of a single attribute. Zhang and Foo [12] suggest that an autonomous agent is equipped with two kinds of control power: one determines its action choice and the other controls its internal state transition. They model autonomy as an action choice function $H: M \rightarrow A$, and a state transition function $T: M \times A \rightarrow M$, where M is the set of all the possible internal states of an agent and A is the set of all actions it can do.

The tasks performed by a system of agents have different characteristics, which lead the corresponding actions to be critical, deducible or primitive. Accordingly, each task achievement ought to be assigned to a different level of autonomy based on the dimensions of an autonomy spectrum as illustrated in Figure 1 [3]. Therefore, an agent performing a task requiring multiple actions may also operate at multiple level of autonomy. It hands over to another capable agent if the decision making exceeds its empowered autonomy level [9].

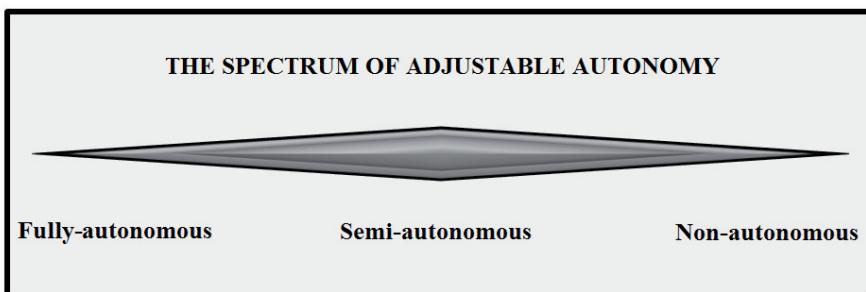


Figure 1. The adjustable autonomy spectrum of an agent (variation from [3])

The degree of autonomy of an agent can be signified by the degree of intervention in its decision making process by interveners [13]. The intervention is an enforcing procedure that an agent is compelled to obey [8]. Giving a command to an agent on how to do, when to do or when not to do a specific action, or blocking some available resources in order to manipulate an agent's behavior are examples of interventions. The main cause of the intervention in the agent decision-making process is its lack of local knowledge or global authority [1], [5], [9]. The intervener can be human or other agents and the intervention can be either within or without the system.

Based on our study, there are two types of interventions: direct and indirect interventions. The direct intervention represents an adjustment in agent decision-making parameters or restricting an agent from accessing specific resources. The indirect intervention denotes an amendment in an agent system's activities or its environmental state or by sharing information from third party that influence the agent's decision [9]. Indirect interventions can be considered as part of the environment dynamism in which the agent persists [10]. Such interventions do not affect an agent's autonomy, but do affect its performance. For instance, the agent might perform wrongly as a result of inputs gained from a defective sensor. This paper only focuses on the direct type of interventions.

3. Related Works in Adjustable Autonomy

Many works in agent-based autonomous systems emphasize that individual agent autonomy design is a critical issue that includes engineering the autonomic relationships prevailing among agents community [1, 2]. Flexible approaches such as the adjustable autonomy is a proven solution to similar cases [8]. Some approaches to agent autonomy adjustment focus on the plans generation part [2] others like KAoS propose that the adjustment should constrain agent behavior [5].

While the adjustable autonomy solves many issues in autonomous systems, it has created other problems. Inyama et al. [4] among others highlight that a key problem of modeling adjustable autonomy is the configuration of a mechanism to ensure the appropriate distribution of autonomy and its adjustment. For example, when an autonomous agent performs a task, the autonomy distribution and the access to the system's resources should be concurrently adjusted during its mission to achieve the mission's goal. Some of the attempts in modeling practical autonomy are summarized as follows.

Barber [3] proposes Sensible agent architecture to demonstrate flexible, receptive and adaptive automated systems. The Sensible model is designed to operate in dynamic environments. The model allows an agent to dynamically configure its autonomy level based on its understanding of the situations. The autonomy configuration for a particular situation is obtained through a predictive process of an autonomy reasoning module. In the Sensible model, the agent reasons about situation handling through two decision-making phases, which are tasks selection and tasks delegation. In the model, determining which agent can decide on the current goal is made via a voting scheme, whereby the agent needs to gain a certain number of votes in order to qualify for decision making [9]. The agent behavior is however, hard to predict as the autonomy configuration is based on the agent's internal states.

The National Aeronautic and Space Administration (NASA) is one of the leading research institutions that experiments with adjustable agent autonomy approach. Adjustable autonomy is used for human-agent groups involving a mix of humans and autonomous agents, namely, Mixed-Initiative adjustable autonomy for human-agent teamwork [5], [14], [15]. The model facilitates the simulation of work situations to support the design of robots and software agents to work in cooperation with humans. They consider scheduled and unscheduled activities, work practice emergence and constraints like resources availability.

Bush et al. [16] utilize task-directed adaptive sensing technology to develop a risk-based adjustable autonomy system. They propose the Autonomous Robot Control via Autonomy Levels (ARCAL) architecture. The ARCAL aim is to stabilize system autonomy operation by an operator at a level with confidence of mission success using gains operator trust concept. Hence, a system is designed with a synergistic and dynamical relationship among the decision makers such that it evaluates its performance and request human assistance whenever needed. The ideas are experimented and tested in natural disaster recovery task simulation.

A survey conducted by Ball and Callaghan [11] discusses four different autonomy management styles in intelligent environments. It reveals people's impression about autonomy in smart homes. The autonomy is organized in four levels: fully-autonomous, semi-autonomous with low and high autonomy, and non-autonomous. The survey results show that different people prefer different levels of autonomy in different situations and for different sub-systems. In time, peoples' views about the autonomy level configuration may drift as a consequence of experience with the technology. The authors conclude that the peoples' attitudes are directed towards adjustable autonomy and they perceive the configuration of the autonomy level based on task performances [10].

4. Ensuring Dynamic Autonomy through the LAA Model

A major challenge in agent-based autonomous system design is the construction of an efficient, continuous and transparent adjustable autonomy of an agent. The design needs to elaborate a mechanism that manages dynamic interaction between the decisions-makers (humans and agents) to achieve performance expectations. The Layered Adjustable Autonomy (LAA) model is proposed to manage some of the autonomy adjustment issues in a MAS, which includes the dynamic autonomy adjustment mechanism and the decision-making management. In the following subsections, we elaborate the LAA model.

4.1. Layered Autonomy

We conceive a novel concept of layered autonomy as follows: Some of the decision-making authorities with regards to a specific event are switched from one agent to a more qualified agent when a threshold of autonomy level of the agent has exceeded. The more qualified agent is the one that has the knowledge and authority to handle an event with greater autonomy. Otherwise, if there are no qualified agents in the system that can make a decision, the system requests for third party intervention [7].

The LAA separates the autonomy into layers that are used to encapsulate an agent's decision process adjustment. Each layer is attributed to deal with actions that correspond to its autonomy level. A qualified agent that fulfills the autonomy threshold at a specific layer is said to be active in that layer. Its autonomy can be adjusted based on the policies and rules associated with the layer. Agents with higher autonomy levels have more responsibilities and higher authority than the agents with lower autonomy levels. Thus, the agents with higher autonomy levels are more likely situated at the top layers of the LAA and the agents with lower autonomy levels act at the lower layers. An illustration to the LAA model is represented in Figure 2.

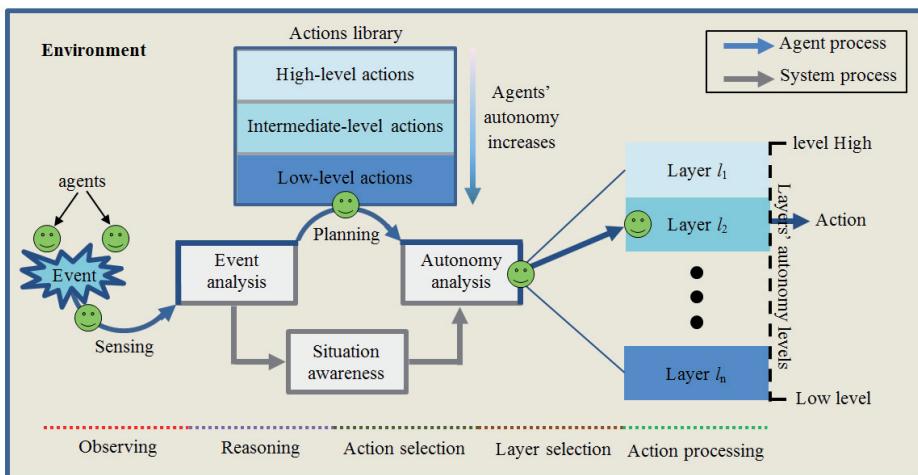


Figure 2. The concept of the LAA

Actions are usually formulated at different levels of abstractions based on the corresponding task criticality [9]. Consequently, we classify the actions of a system based on its functionalities into three types: high-level (critical), intermediate-level (deducible) and low-level (primitive) actions. In this setting, we assume that each level of actions is assigned to its specific autonomy layer as shown in Figure 2 and each actor in the system (e.g. human or agent) has a set of possible actions along with their autonomy properties. Some of the autonomy properties of a layer are different from that of the other layers, such as the adjustment mechanism of actions' parameters (e.g. popping up an interface dialog box for requesting feedback from human).

To illustrate the above, we give an example of four people A, B, C and T. Let us say A, B, C saw a thief, T, trying to rob a house (an event). Each of the three selects an action from the possible actions that they have, which are as follows: A calls the police (primitive action), B shouts at T (deducible action), C tries to catch T (critical action).

As we can see, the three actions are at different levels of complexity and there is a conflict between the actions as well. If B shouted at T, then C will have difficulty catching T as the shout scares the thief to flee. In this scenario, however, all the three seems to have knowledge on how to act during the event by selecting a convenient action based on their abilities while each of their authority and responsibility differs.

To determine who among the three should decide requires additional information. An Autonomy Analysis Module (AAM) and Situation Awareness Module (SAM) control the distribution of the autonomy to an actor based on the situation. Hence, in the LAA model, the higher authority agent is the one who makes the decision (i.e. by default). Nevertheless, in some circumstances, imposed constraints might direct a less authorized agent to act.

4.2. The LAA Modules

The LAA model utilizes two modules in the autonomy analysis and distribution processes which are an Autonomy Analysis Module (AAM) and Situation Awareness Module (SAM). The AAM is a management module that handles dynamic autonomy distribution. In a given situation, the autonomy analysis module is responsible for ensuring that the agent's autonomy level and the layer are compatible with the selected action. Preconditions and post-conditions of each must be fulfilled during the autonomy analysis [7].

The SAM is a reasoning method that is utilized via the system to assist adjustable autonomy planning process. The SAM provides information about the system's current situation along with the external influences, reasons the situation and derive decisions to direct the system to counter the situation. In its reasoning, it relies on three main functions which are situation perception function, situation comprehension function and situation projection function [17]. It works with the AAM to determine the selected actions' validity in which the desired situation is met (see Figure 2).

4.3. Logical Representation of the LAA Model

This section presents the logical representation of the Layered Adjustable Autonomy (LAA) model. We assume that the agent has the characteristics of autonomy, reactivity, pro-activity, goal-directedness, embodiments, sociality, along with the intelligence and the rationality [18]. Let En be a dynamic environment that has a set of discrete instantaneous states at time, t , which is an increasing period of time:

$$En = \{Sen_1, Sen_2, \dots\}$$

Let Ev be the set of all possible events that can occur in the environment En :

$$Ev = \{ev_1, ev_2, \dots\}$$

Let Ag be the agents that are situated in En , whose corresponding internal (mental) states at time t are captured by SAG :

$$Ag = \{ag_1, ag_2, \dots\}$$

$$SAG = \{SAG_1, SAG_2, \dots\}$$

The agents Ag have a set of possible actions Ac that they can perform:

$$Ac = \{ac_1, ac_2, \dots\}$$

A sequence of actions performance in the environment that leads to the achievement of a goal represents a mission processing Run ; in which, run represents the execution of a selected action during specific environmental state by an agent using an action execution function Exe .

$$Run = \{run_1, run_2, \dots\}$$

$$Exe : Sen_1 \times ac \rightarrow Sen_2$$

Although Run represents the history of the goal achievement, it cannot be a linear process in a dynamic environment. Therefore, it is unrealistic to configure the required Run (executable plan) during planning stage. Instead, a preliminary plan configuration that shows the main task of the mission is more practical approach [16]. For example an agent reasoning about run_2 in a history dependent model might be influenced by run_1 .

Consequently, we illustrate the logic of the LAA model through providing two case-based scenarios as follows:

Case 1: An agent, ag_1 , makes a decision of selecting action ac_1 based on its reasoning over its internal state Sag_1 , and the environmental state Sen_1 , executes the ac_1 and changes the environmental state Sen_1 to Sen_2 [18].

Case 2: An agent, ag_1 , makes a decision of selecting an action ac_1 , based on its reasoning over its internal state Sag_1 , and the environmental state Sen_1 and submits to the LAA procedure in order to perform the ac_1 .

In the traditional case (Case 1), an agent's design objectives are characterised by functions such as Observation Obs , whose output is the agent's perception, Per ; state Transition, Tra whose output is the agent's updated internal state; and action choice Selection, Sel whose output is the selected action. These functions control the agent's behaviour [12, 18].

Assume that in En , there exists an agent $ag_1 \in Ag$, with internal state $Sag_1 \in SAg$, and handles an event $ev_1 \in Ev$ in the current environmental state Sen at time, $t=1$, Sen_1 . We can treat ev_1 as a function that changes En by executing an action. The following propositions describe Case 1 scenario in which ag_1 responds to ev_1 :

$$\text{Case 1} = \langle En, Sen_1, ev_1, ag_1, Sag_1, Obs, Tra, Sel, Exe, Ac, ac_1, Per \rangle$$

$ev_1 : En \rightarrow Sen_1$	(An event occurs in the environment)
$Obs : Sen_1 \rightarrow Per_1$	(An agent perceives the change)
$Tra : Sag_1 \times Per_1 \rightarrow Sag_{1+Per_1}$	(The agent updates its internal state)
$Sel : Sag_{1+Per_1} \times Ac \rightarrow ac_1$	(The agent selects an action)
$Exe : Sag_{1+Per_1} \times Sen_1 \times ac_1 \rightarrow Sen_2$	(The action changes the environment)

where Sen_2 represents the change to the environment due to ag_1 reacting on ev_1 , based on its updated internal state Sag_{1+PerI} and the selected action ac_1 .

In an interactive system that belongs to a dynamic environment, the scenario captured by Case 1 is impractical as the agent decision is made based on its imperfect perception while the system and its surrounding are continuously changing. In addition, there are no measures to help promote an agent to react to an event, which can ensure a more certain result. This led to the conception of the LAA model that incorporates mechanisms to monitor system internal states and the global environmental state to dynamically coordinate system autonomy distribution as the autonomy adjustment is the key element of an agent discipline. Consequently, Case 2 addresses the limitations of Case 1, and is presented as follows:

Firstly, the actions in the LAA model are further categorized into high-, intermediate- and low-level actions, that correspond to non-autonomous \overline{Ac} , semi-autonomous \widetilde{Ac} and fully-autonomous actions \dot{Ac} .

$$Ac = \{\overline{Ac}, \widetilde{Ac}, \dot{Ac}\}$$

where, $\overline{Ac} = \{\overline{ac}_1, \overline{ac}_2, \dots\}$, $\widetilde{Ac} = \{\widetilde{ac}_1, \widetilde{ac}_2, \dots\}$ and $\dot{Ac} = \{ac_1, ac_2, \dots\}$.

Secondly, the LAA model provides an Autonomy Analysis Module (AAM) to ensure dynamic autonomy distribution capability. The AAM, Φ , through an autonomy function Aut , identifies which of the agents, Ag , that pursues a semi-autonomous action, say \widetilde{ac}_1 , has the knowledge (*know*) and authority (*can*) to implement the action \widetilde{ac}_1 . The agent then selects an autonomy layer, $l_i \in L$, such that l_i provides the required resources $r_i \in R$ including the adjustment process when necessary. We assume that the *know* and the *can* conditions of an agent are known as their formation is a domain specific issue.

$$L = \{l_1, l_2, \dots\}$$

$$R = \{r_1, r_2, \dots\}$$

Thirdly, the pursued action, say \widetilde{ac}_1 , has a set of corresponding fluents $f_{l_{1,j}} \in Fl$, which are predicate conditions $pr_{i,j} \in Pr$ with boolean values that reason about the action based on time argument and have a function $fn_{i,j} \in Fn$, or more of such, each of which takes inputs, $(x_1, y_1, \dots) \in I_j$ and produces outputs, O_j [19, 20]. The first fluent, $f_{l_{1,0}}$ of each action holds an *autonomy update function* $fn_{i,0}$. The Φ modifies the parameters of the $fn_{i,0}$ to set the actions autonomy level instructions to the corresponding agent.

$$Fn = \{fn_{i,1}, fn_{i,2}, \dots\}$$

$$Fl = \{f_{l_{1,1}}, f_{l_{1,2}}, \dots\}$$

$$f_{l_{1,1}} : pr_{i,1} \rightarrow \{\top, \perp\}$$

$$Pr_{i,1}(fn_{i,1}(x_1, y_1), t)$$

$$fn_{i,1} : I_1 \rightarrow O_1$$

$$\widetilde{ac}_1 = \{fl_{1,0}, fl_{1,1}, fl_{1,2}, \dots\}$$

$$\widetilde{ac}_1 = \{pr_{1,0}(\text{fn}_{1,0}(x_0, y_0), t), pr_{1,1}(\text{fn}_{1,1}(x_1, y_1), t), pr_{1,2}(\text{fn}_{1,2}(x_2, y_2), t), \dots\}$$

The global direction of the system is determined by a Situation Awareness Module that is concerned with determining the situation awareness state configuration, $Ssit$, to support the Φ control. The following propositions describe Case 2 scenario in which an agent ag_1 responds to ev_1 that occurs in En following the LAA model:

Case 2 = $<En, Sen_1, ev_1, ag_1, Sag_1, Obs, Tra, Sel, Exe, \Phi, Ac, \widetilde{ac}_1, l_2, r_2, Per, Aut, Ssit>$	
$ev_1 : En \rightarrow Sen_1$	(An event occurs in the environment)
$Obs : Sen_1 \rightarrow Per_1$	(An agent perceives the change)
$Tra : Sag_1 \times Per_1 \rightarrow Sag_{1+Per_1}$	(The agent updates its internal state)
$Sel : Sag_{1+Per_1} \times Ac \rightarrow \widetilde{ac}_1$	(The agent selects an action)
$\Phi : Sag_{1+Per_1} \times Ssit \rightarrow \Phi_t$	(The AAM updates itself)
$\Phi_t : ag_1 \Leftrightarrow [Aut((know(\widetilde{ac}_1, k_1), can(\widetilde{ac}_1, c_1), t))]$	(The AAM checks the autonomy conditions for the agent and the action)
$\Phi_t : [(ag_1 \times \widetilde{ac}_1) \leftarrow (pr_{1,0}(\text{fn}_{1,0}(x_0, y_0), t))]$	(The AAM distribute the autonomy)
$ag_1 : \Phi_t \times \widetilde{ac}_1 \rightarrow l_2$	(The agent select appropriate layer)
$l_2 : r_2 \rightarrow ag_1$	(The layer provides resources to the agent)
$ag_1 : Sag_{1+Per_1} \times r_2 \times Sen_1 \times \widetilde{ac}_1 \rightarrow (Exe (fl_{1,0}, fl_{1,1}, fl_{1,2}, \dots)) \rightarrow Sen_2$	(The agent execute the action and the action changes the environment)

where k_1 is the *know* value provided by the corresponding agent ag_1 and c_1 is the *can* value determined by the system and assigned to the corresponding agents with regards to the pursued action, say \widetilde{ac}_1 .

The LAA provides a mechanism that deals with concurrently graded autonomy. Consequently, checking an agent's ability to perform an action causes a delay that precludes the need of an intervention. The continuous checking of all aspects of the environment and the agents enables dynamic coordination of the system, which enables the agents to effectively perform under different states of constraints.

Similarly, when pursuing non-autonomous, \overline{ac}_1 and fully-autonomous, ac_1 actions, different autonomy distribution, layers and resources need to be arranged to support the agent actions. If for instance, three agents: ag_1 , ag_2 and ag_3 react to ev_1 by selecting different actions, say ag_1 and ag_2 select \widetilde{ac}_1 while ag_3 selects ac_2 , then the following proposition describes the autonomy distribution by the AAM at time t :

$$\Phi_t : \text{select}(ag_1, ag_2, ag_3) \Leftrightarrow [Aut(know((\widetilde{ac}_1, k_1, k_2) \vee (ac_2, k_3)), can((\widetilde{ac}_1, c_1, c_2) \vee (ac_2, c_3)), t)]$$

Wooldridge [18] proposes that the agent can select an action and performs a decision regarding the selected action based on the system's history that the agent witnessed. System's history includes logistical information such as work-load reports of the agents, which can be utilized to distribute the autonomy among the agents [2]. As a result, an agent acquires new knowledge to make a better decision by observation and learning (internal behaviour). The authority of the agent is gained and enhanced as a consequence of the knowledge gained and successful decision-making (external behaviour).

5. Conclusion and Future Work

The dynamic autonomy adjustment is still a challenging task and is clearly stated as a domain specific issue. Among many applications, the criteria for autonomy adjustment mechanism differ. The proposed Layered Adjustable Autonomy (LAA) as a model outlines an idea for managing adjustable autonomy in the MAS using layers of autonomy levels, an Autonomy Analysis Module (AAM) and Situation Awareness Module (SAM).

In the proposed model, we restrict the conditions that qualify the agent to make a decision by its knowledge and authority to perform an action. Agent's knowledge represents its ability to understand an event and select an appropriate action towards the event while the authority of the agent is decided based on specific circumstances. Consequently, the criteria for autonomy dynamic distribution are configured via the AAM and SAM based on the selected action type and situation complexity. The situation awareness state might result in a less authorized agent being selected to act due to some circumstances or constraints. The model provides flexibility of selecting achievable actions by categorizing the actions according to their complexity and accordingly fixing the autonomy range of each to satisfy the situation.

In our future work, a formulation to autonomy measurements for the LAA model will be proposed. Furthermore, a mechanism with situation awareness capability is to be proposed to assist in system autonomy distribution along with the autonomy formula.

References

- [1] J. Anderson and M. Evans, Supporting Flexible Autonomy in a Simulation Environment for Intelligent Agent Designs, *In the Proceedings of the Fourth Annual Conference on AI, Simulation, and Planning in High Autonomy Systems*, IEEE, Tucson, Arizona (1993), 60–66.
- [2] E. Mark, J. Anderson and G. Crysdale, Achieving Flexible Autonomy in Multi-Agent Systems using Constraints, *Applied Artificial Intelligence: An International Journal* **6** (1992), 103–126.
- [3] K.S. Barber, The Architecture for Sensible Agents, *In Proceedings of the International Multidisciplinary Conference, Intelligent Systems: A Semiotic Perspective*, CiteSeerX, Gaithersburg, MD: National Institute of Standards and Technology (1996), 49–54.
- [4] H.C. Inyama, S.U. Ufoaroh and V.C. Chijindu, Intelligent Agents Autonomy Issues, *Afr J. of Comp & ICTs* **5** (2012), 49–52.
- [5] J.M. Bradshaw, P.J. Feltovich, H. Jung, S. Kulkarni, W. Taysom and A. Uszok, Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction, *In Agents and Computational Autonomy: Potential, Risks, and Solutions*. In: M. Nickles, M. Rovatos and G. Weiss, (eds.), *LNAI*, Springer, Berlin/Heidelberg (2004), 17–39.
- [6] R. Falcone and C. Castelfranchi, Levels of Delegation and Levels of Adoption as the Basis for Adjustable Autonomy, *Advances in Artificial Intelligence. LNCS*, **1792** (2000), 273–284.
- [7] S.A. Mostafa, M.S. Ahmad, M. Annamalai, A. Ahmad and S.S. Gunasekaran, A Conceptual Model of Layered Adjustable Autonomy, *In the 2013 World Conference on Information Systems and Technologies (WorldCIST'13)*, Springer, Algarve, Portugal (2013).
- [8] M.J. Huber, Considerations for Flexible Autonomy within BDI Intelligent Agent Architectures, *In the 1999 AAAI Spring Symposium, in Agents with Adjustable Autonomy*, Technical Report SS-99-06 (1999), 65–72.
- [9] K.S. Barber, A. Goel and C.E. Martin, The Motivation for Dynamic Adaptive Autonomy in Agent-based Systems, *In Intelligent Agent Technology: Systems, Methodologies, and Tools. Proceedings of the 1st Asia-Pacific Conference on IAT*, Hong Kong, eds. J. Liu and N. Zhong, World Scientific, Singapore, CiteSeerX (1999), 131–140.
- [10] M. Ball and V. Callaghan, Managing Control, Convenience and Autonomy: A Study of Agent Autonomy in Intelligent Environments, *In Special Issue on Agent-Based Approaches to Ambient Intelligence, AISE series*, IOS Press (2012), 159–196.

- [11] M. Ball and V. Callaghan, Introducing Intelligent Environments, Agents and Autonomy to Users, *In the Seventh International Conference on Intelligent Environments (IE)*, IEEE Press, Nottingham (2011), 382–385.
- [12] D. Zhang and N. Foo, Autonomy of Autonomous Agents, *PRICAI 2000, LNAI*, Springer-Verlag, Berlin Heidelberg **1886** (2000), 318–328.
- [13] K.S. Barber, D. Han and T.H. Lui, Strategy Selection-based Meta-level Reasoning for Multi-agent Problem-solving, *In: P. Ciancarini and M.J. Wooldridge (eds.): Agent-Oriented Software Engineering: AOSE 2000. LNCS 1957* (2001), 155–187.
- [14] J.M. Bradshaw, M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok and R. Van Hoof, Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications. *In H. Hexmoor, R. Falcone and C. Castelfranchi, (eds.) Agent Autonomy. CiteSeerX*, Kluwer (2003), 243–280.
- [15] D. Dumond, J. Ayers, N. Schurr, A. Carlin, D. Burke and J. Rousseau, Coordinating with Humans by Adjustable-Autonomy for Multi-robot Pursuit (CHAMP), Unmanned Systems Technology XIV, *In: Karlsen, E. Robert, Gage, W. Douglas, Shoemaker, M. Charles, Gerhart and R. Grant, (eds.) Proceedings of the SPIE, the Smithsonian/NASA Astrophysics Data System* **8387** (2012), 838703–838703-15.
- [16] L.A.M. Bush, A.J. Wang and B.C. Williams, Risk-Based Sensing in Support of Adjustable Autonomy, *Aerospace Conference, IEEE Press*, Big Sky, MT (2012), 1–18.
- [17] Y. Lili, Z. Rubo and G. Hengwen, Situation Reasoning for an Adjustable Autonomy System, *International Journal of Intelligent Computing and Cybernetics* **5** (2012), 226–238.
- [18] M.J. Wooldridge, *An Introduction to Multi-agent System* (2nd ed.), A John Wiley and Sons Ltd, Torquay, UK, 2009.
- [19] F. Pirri and R. Reiter, Some Contributions to the Meta-theory of the Situation Calculus, *J.ACM* **3** (1999), 325–361.
- [20] F. Lin, Situation Calculus. *In F. Van Harmelen, V. Lifschitz and B. Porter (eds.), chapter 16, Handbook of Knowledge Representation*, Elsevier (2008).

K-depth RDF Keyword Search Algorithm Based on Structure Indexing

Minho BAE^{a,1}, Duc NGUYEN^a, Sanggil KANG^b and Sangyoon OH^a

^a Department of Computer Engineering, Ajou University,
Suwon, Rep. of Korea

^b Department of Computer and Information Engineering, Inha University,
Incheon, Rep. of Korea

Abstract. Information retrieval from large scale RDF datasets is a challenging task. Because it takes much time to process query and it is hard to store a large collection of RDFs, it requires an efficient method to index and query. In this paper, we propose a novel RDF management system architecture along with indexing and querying algorithms. Our empirical experiments show our system performs substantially better than the conventional system. Also we verify the effectiveness of k-depth concept in our design with additional experiments.

Keywords. RDF, Structure Index, Keyword Search, K-depth

1. Introduction

Resource Description Framework (RDF) language that is proposed by Tim Berners-Lee provides a way to represent semantic information for various domains. Specially, RDF can be used to describe resources in World Wide Web including web pages and diverse applications can share information annotated by the common language, RDF.

RDF data collection consists of a set of subject-predicate-object (SPO) triples. The SPO triple is a pair of entities with a named relationship (i.e. connection between entities) or an entity connected to the value of a named attribute from the entity-relation (from the ER perspective). We also can view RDF data as a graph of typed nodes and edges where nodes correspond to entities and edges to relationships. An object of a given triple can be a subject of other triples [1]. An example of RDF data is presented in Table 1 as well as is depicted in the graph format as seen in Figure 1.

Managing (i.e. storing and retrieving) a billions of triples such as linked data is a challenging issue in both research communities and commercial companies. Recently there are a lot of proposals about indexing and querying huge size of RDF collections. One of major approaches to query RDF data is using conjunction of triple patterns. In this approach, a triple pattern is a triple with variables and the same variable in different patterns denotes a join condition [2]. For example, if we want to find “Jon Foobar” from the RDF data collection, we can express it as <?x foaf:name Jon Foobar>. It is very expressive but very restrictive. There are many popular RDF managing systems providing a storage infrastructure for RDF data. To use this

¹ Corresponding Author: Minho Bae, Paldal 908, Ajou University, Suwon, Rep. of Korea; E-mail: minkkang@ajou.ac.kr

approach, users must be familiar with the structure of the current RDF document as well as a structured query language such as SPARQL [3]. Thus a new approach that enables users to search RDF graphs using keywords is required to increase the usability of such data sources. Also, it gets more popular recently.

However, the most of current systems are not suitable to support typical querying scenarios such as keyword searching for Web data. Also they are not performing well either in response time [4]. In this paper, we propose a novel indexing and querying approach to support typical keyword searching over the Web data in RDF. Also we design a system that supports our approach. In the experimental section, our system shows the improvement with respect to query answering performance and capabilities compared to current RDF storage systems.

Table 1. RDF data example

Subject	Predicate	Object
id123456	foaf : name	Jon Foobar
id123456	rdf : type	foaf : Agent
id123456	foaf : weblog	http://foobar.xx/blog
http://foobar.xx/blog.rdf	rdf : type	rss:channel
http://foobar.xx/blog.rdf	foaf : maker	Id123456
http://foobar.xx/blog	rdfs:seeAlso	http://foobar.xx/blog.rdf
http://foobar.xx/blog	dc : title	Title

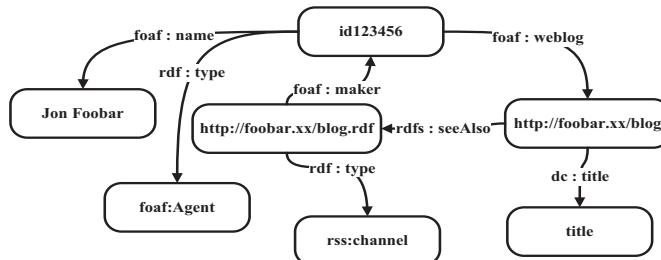


Figure 1. RDF data example of Table 1 in graph format

2. Related Works

In Ref. [5], an algorithm for top-k exploration of sub-graphs to retrieve the top-k most relevant structured queries is proposed. In their proposal, keyword index is used for elements and structure index for RDF data graph. The keywords in the query are translated into expressive formal queries. First, instead of mapping the keywords to data tuples, they are mapped to the elements in the data graph using keyword index. Next, with each mapping of such keyword elements, the summary graph (i.e. structure index of the original RDF data graph) is explored to search for the augmented query graph (substructure) connecting them. From these sub-graphs, conjunctive queries are created by mapping the edges with predicates and the vertices with variables or literal values. Then their system generates the top-k queries by a scoring function, instead of

computing answers for the keywords. Finally, users need to select their appropriate queries from these proposed structure queries to find the necessary information with their needs. The approach has many interesting capabilities. However, possible information loss occurs because users customize with small value of k parameter is one of drawbacks. In Ref. [6], they improve previous works with respect to accuracy of query results by introducing a new phase: the new phase transforms a sub-graph query to a set of entities by traversing the sub graph. However, the new approach requires more time to process while it improves the accuracy of results.

In Ref. [2], the keyword search system over structured data directly retrieves results for the keyword query. From a phrase of keywords, they use a sub-graph retrieval algorithm to return a set of sub-graphs that match with the query keywords by their ranking model. To do so, for every triple data, they maintain the list of keywords derived from the subject and object associated predicate. Then they create an inverted index for each keyword query with a list of corresponding triples so that they can join these data to get sub-graphs of all keywords in the query by adapting the backtracking algorithm [7]. Considering and maintaining keywords from all 3 factors (subject, predicate, and object) really needs time, memory and well-designed schema rather than from the objects' literal values only.

Even though the state of art in RDF management system support users' request well, there are performance and scalability problem to be addressed. First, the querying performance should be improved. Since RDF is a graph format, querying is very slow. Also, the current graph or relational data based storage cannot hold large scale RDF data.

3. K-depth RDF Keyword Search System

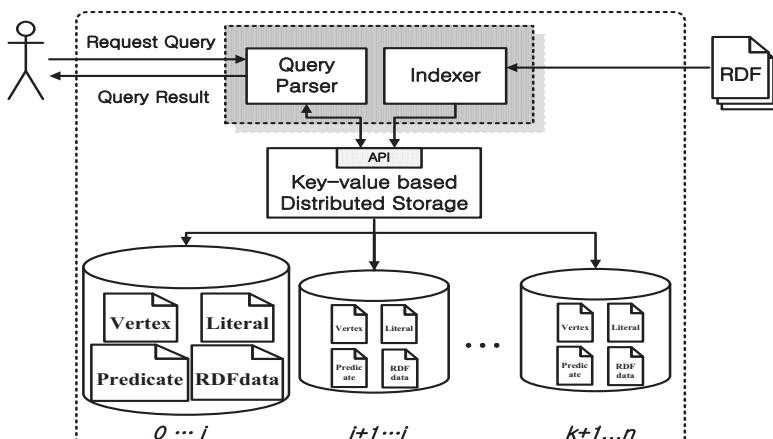


Figure 2. RDF Keyword System Architecture

In this chapter, we describe our proposed system architecture for our RDF keyword search algorithm based on structure index as seen in Figure 2.

For our system, we assume that for keyword search query results users prefer to search objects i.e. literals that describe the properties or characteristics than subjects that identify the objects.

To address the size issue, we adopt a key-value storage design because the key-value storage is capable of loading the values that stored in other nodes quickly [8] and can be distributed to support big-size RDF collections. In the following subsection, we describe our indexing design including data structure and querying algorithm.

3.1. Data Storage Structure

In our system, there are four collections which are the collections of nodes and relations of given RDFs: 1) vertex collection that stores subjects and objects (since a subject can be an object and vice versa in RDF), 2) predicate collection that stores predicate data, 3) literal collection that stores literal data, and 4) RDFdata collection that stores relations of data. RDFdata collection is the most important among four collections because it is the first indexing product of our algorithm and other three collections are derived from this. We do not store data in triples like conventional RDF infrastructure (e.g. Jena [9]).

3.1.1. Collections

RDFdata collection has *Vertices* documents i.e. entities in a collection that represents each value i.e. object or subject, and *Up & Down* document that represent data in k-depth one. k-depth is a degree that shows the number of vertices documents that are related to the current vertices document. When the vertices document is an object, *Up* document stores data group that has a predicate as a key and subjects as values e.g. {“predicate_1”: [subject 1, subject 2,...,subject n]},..., {“predicate_n”: [subject n+1, subject n+2,...,subject m]}. On the other hands, when vertices document value is a subject, *Down* document stores objects e.g. [object 1, object 2,...,object n]. We do not store predicates in down document because we are able to deduce its predicate through its up document. For example, if we have “<http://foobar.xx/blog.rdf>” as vertices document which is both a subject and an object at the same time, we have *Up & Down* documents. If the stored vertices document in RDFdata collection is only used as subjects, there is only stored in down. If the stored vertices document is only used as objects, there is only stored in up. The main reason of this design is to improve the performance of information retrieving through storing related data as grouped together. Table 2 shows an example of RDFdata collection in Table 1. In Table 2, elements are stored as numbers, *Flags*. The flag is explained in detail in 3.1.2.

Table 2. RDFdata collection from Table 1

Vertices	Up	Down
[0]	{“2”: [3]}	[1,2,5]
[1]	{“0”: [0]}	[]
[2]	{“1”: [0]}	[]
[3]	{“4”: [5]}	[0,4]
[4]	{“1”: [3]}	[]
[5]	{“3”: [0]}	[3,6]
[6]	{“5”: [5]}	[]

3.1.2. Indexing

As mentioned, our RDF management system and querying algorithm are utilizing four collections of given RDF data. Since the performance to retrieve data from RDF

documents gets slower as we have more RDF data to store, we need an effective method to reduce the size of stored data. We propose a novel indexing algorithm with three steps to optimize storage space.

- Flagging

It is more space-efficient to translate strings(vertex/literal) into numbers and store the numbers instead of native strings to the storage. Thus we use a flagging technique (*Flag*). To apply this, we introduce predicate collection and vertex collection as described. Values in those two collections have corresponding string objects. pID and vID are assigned to predicate values and vertex values respectively in the collections.

In general, literal type data is used in object, not in subject of RDFs. Since we assume that the keywords used in users' information search are literal, we build literal collection as shown in Table 3 with literal data to expedite the search process. We identify literal from RDFs using Jena [9]. Table 3 shows three collections from example in Table 1.

Table 3. Three collections from Table 1

Vertex collection		Predicate collection		Literal collection	
vID	Value	pID	Value	ID	Value
0	id123456	0	foaf : name	1	[Jon, Foobar]
1	Jon Foobar	1	rdf : type	6	[title]
2	foaf : Agent	2	foaf : maker		
3	http://foobar.xx/blog.rdf	3	foaf : weblog		
4	rss : channel	4	rdfs : seeAlso		
5	http://foobar.xx/blog	5	dc : title		
6	title				

- Stemming

As noted, we build a literal collection to expedite the search process. To improve efficiency in space and accuracy, we store literal data using stemming. Stemming is the process of bringing the inflected words or derivatives into their original form (stem or root). It is useful for determining the mapping of different words with same meaning, even the case in which the root does not have a valid semantic e.g. the word "compiler" and "compilation" are stemmed to "compil". They are considered to be related. Stemming is used popularly in many search engines and the problem of natural language processing. One of the most popular programs of stemming is the algorithm provided by Martin Potter [10].

- Grouping

To reduce storage space further, we adopt a way of grouping all same vertices by checking the *UP & DOWN* list. The rows that have same *UP & DOWN* lists will be grouped together into a single row. We name this procedure as grouping. Table 4 shows the grouping of the given example

Table 4. Grouping example

Vertices	Up	Down
[0]	{"1": [2,3], "2": [4]}	[5]
[1]	{"1": [2,3], "2": [4]}	[5]

↓

Vertices	Up	Down
[0,1]	{"1": [2,3], "2": [4]}	[5]

3.2. Search Algorithm (Querying)

- K-depth

In our system, we introduce a metric k-depth that describes the relevancy of the user-searching information with the query results. K-depth represents the number of relation between nodes. For example, “<http://foobar.xx/blog.rdf>” has k-depth one with “id123456” and k-depth two with “Jon Foobar”.

- Search algorithm

To use our proposed system for keyword search over RDFs, a user inputs keywords and k-depth as a start. As noted, we assume that users search information literal type keywords. Thus, our system first lookups the literal collection table for the given keywords. If the keywords are found, it takes the corresponding IDs. Then it retrieves vertices data with the same IDs from RDFdata collection i.e. *Up & Down* documents. The vertices data will be stored at the temporal storage in the memory, **R**. The results are discarded if the value (i.e. vertices) already exists in the **R**. We iterate the algorithm as decreasing k-depth value as one. The algorithm terminates when we have k-depth one and the results in **R** are returned to the user. Figure 3 depicts the algorithm running with the example in Table 1.

Keyword : title, K-depth :3

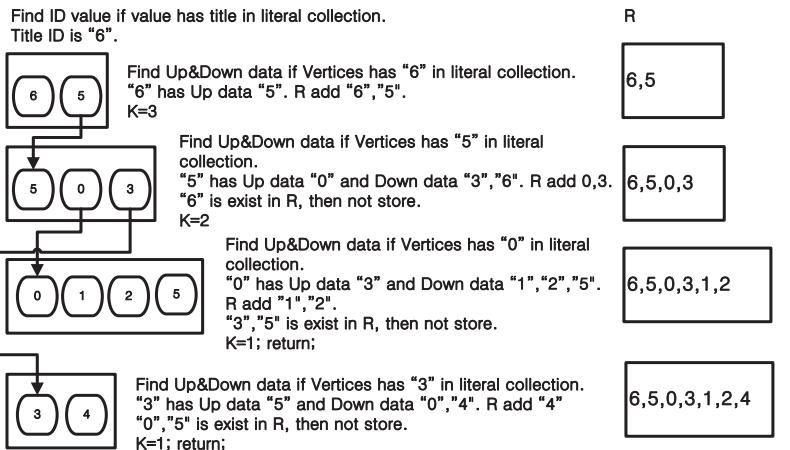


Figure 3. SearchGraph algorithm example

The following pseudo code describe *SearchGraph* algorithm that is iterated recursively, until the k-depth becomes one.

Algorithm	SearchGraph(k, vID, callerID)
Input :	k, vID, callerID
Data	RDFdata collection (table) with UP & DOWN list
Output :	R containing the triples of k-depth subgraph of vertex
1:	If (k == 0) then
2:	return nothing
3:	Else
4:	R.add(extractTriples(GetUpList(vID)) //Retrieve 1-depth graph of incoming edges of the vertex //(except for triples in which: OBJECT is vID, SUBJECT is callerID)
5:	R.add(extractTriples(GetDownList(vID)) //Retrieve 1-depth graph of outgoing edges of the vertex //(except for triples in which: SUBJECT is calledID, OBJECT is vID)
6:	If (k == 1) then
7:	return result
8:	Else
9:	For each subject in vID.upList
10:	R.add(searchGraph(k-1, subject.ID, vID)) //Add (k-1)-depth graph of adjacent vertices of given vertex
11:	For each object in vID.downList
12:	R.add(searchGraph(k-1, object.ID, vID)) //Add (k-1)-depth graph of adjacent vertices of given vertex

4. Evaluation

In this chapter, we present our empirical evaluation of our proposed system and algorithm. We conduct two experiments for evaluation: we compare querying performance with a conventional RDF management system and conduct k-depth query performance evaluation on two different RDF sets i.e. complex and simple.

4.1. Evaluation Environment Setup

For our evaluation, we use MongoDB [11] as a key-value based distributed storage and utilize its sharding feature to store RDFs as distributed. We have one master node for indexing, querying, and storing and one additional node for data distribution (i.e. sharding). Table 5 shows the environment setup for evaluation.

We build the dataset used for our evaluation with RDF data from DBpedia [12]. We merge three RDF datasets (persondata, short abstracts, and redirects) to build complex sets, since individual datasets have only one k-depth.

Table 5. Simulation Environment Setup

	Main Server	Node
Server Name	WISE	MAGI
CPU clock	1.6 Ghz	2.0 Ghz
RAM	8GB	4GB
Number of core	2	1
OS	Fedora 13	CentOS 5.6
Development	Java 1.7.0	Java 1.7.0
Database	MongoDB	MongoDB

4.2. Keyword Querying Performance

We compare keyword querying performance between two systems: Jena with SPARQL (i.e. a conventional system) and our proposed system. Since Jena with SPARQL does not have the k-depth, we set both queries with k-depth one. Table 6 shows the query response time for datasets with different size. Our system performs 15~27 times better with given datasets.

Table 6. Compared Jena

System	Data Size(M)	1	5	10
Jena	1906.1(ms)	1950.4(ms)	1980.4(ms)	
Our System	69.2(ms)	86.8(ms)	126.6(ms)	

4.3. Effectiveness of K-depth

In this experiment, we verify the effectiveness of our k-depth feature. Since it is a unique feature of our system that isn't supported by any other RDF management system, we conduct empirical evaluation of k-depth effectiveness by varying the query requirements over different complex of RDFs.

First, we conduct the query response performance of our system for various k-depths over complex and simple RDF data sets to show validity of k-depth introduction. In this experiment, we define two relations: the complex relation whereas keywords that we query have many relations to it and the simple relation whereas keywords have not many relations. Thus, there is more data with high k-depth in complex relation datasets than simple relation datasets. Also, it takes more time to process query over complex relation datasets than simple relation datasets with the same k-depth. In Figure 4, we verify the proposition: when we query with higher k-depth, it takes more. Also queries over complex relation datasets takes more processing time.

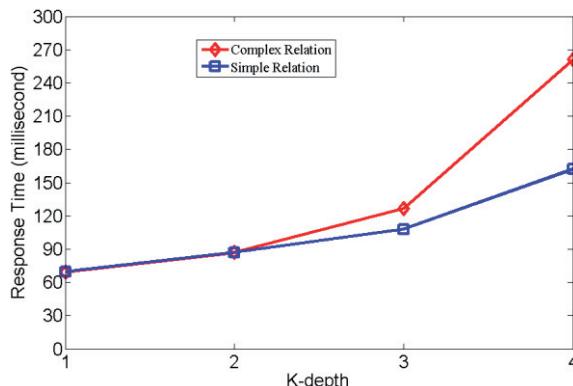


Figure 4. Comparison of query processing time of k-depth test results with complex relation datasets and simple relation datasets

Table 7. Comparison of query processing time of k-depth test results with various sizes of datasets

System	Data Size(M)	10	50	100
Our System		0.1(s)	1.9(s)	9.9(s)

Additionally, we measure the query processing time of our system by varying the size of data and the results with k-depth two over various data size are shown in Table 7. As the size of data for queries gets bigger, it takes longer to process. It is mainly because of memory space limitation. When we are running out of space on memory, we have more I/O to hard disk to read data. Thus overall query performance is degraded. This problem can be resolved by increasing the number of nodes for data distribution.

5. Conclusion

To retrieve information from RDF data, conventional querying methods provide expressive approach. However, there are some drawbacks such as user familiarity to

the structure of the RDF document as well as a structured query language such as SPARQL. Also, it usually takes much longer to process the query. In this paper, we investigate a novel RDF management system architecture to address the problem; in which the system indexes RDF data with our new indexing algorithm with K-depth and process queries over distributed key-value based storages.

We compare our system with the conventional RDF management system (i.e. Jena and SPARQL), and as expected, empirical results shows substantial performance gains by adapting our approach. As well, we verify the effectiveness of k-depth concept with additional experiments.

We expect that the indexing and querying performance as well as storage efficiency of our system can be improved in few points; indexing of RDF data can be parallelized using MapReduce and indexing performance can be improved, since our design is based on key-value based storage system. Also, instead of getting k-depth from users, we decide an optimal k-depth for query by adopting artificial intelligence techniques.

References

- [1] S. Elbassuoni, M. Ramanath, R. Schenkel, G. Weikum, "Searching RDF Graphs with SPARQL and Keywords", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2010.
- [2] S. Elbassuoni, R. Blanco, "Keyword Search over RDF Graphs", Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pages 237-242.
- [3] SPARQL : <http://www.w3.org/TR/rdf-sparql-query/>
- [4] A. Harth, S. Decker, "Optimized index structures for Querying RDF from the Web", Web Congress, 2005.
- [5] T. Tran et al. "Top-k exploration of query graph candidates for efficient keyword search on rdf." Proceedings of the 25th International Conference on Data Engineering (ICDE 2009), Shanghai, China (May 2009). 2009.
- [6] C.S. Nikolaou, "Keyword Search in RDF Databases." (2010).
- [7] S. Wernicke, "A faster algorithm for detecting network motifs.", Algorithms in Bioinformatics (2005): 165-177.
- [8] A. Termehchy, M. Winslett, "Keyword Search over Key-Value Stores", WWW' 10 Proceedings of the 19th international conference on World Wide Web, 2010, Pages 1193-119
- [9] J.J. Carroll, et al. "Jena: implementing the semantic web recommendations." Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM, 2004.
- [10] M.F. Porter, "An algorithm for suffix stripping." Program: electronic library and information systems 40, no. 3 (2006): 211-218.
- [11] K. Chodorow, and M. Dirolf, "MongoDB: the definitive guide", O'Reilly Media, 2010.
- [12] DBpedia : <http://dbpedia.org>.

An Efficient Method for Hiding High Utility Itemsets

Bay VO^{a,1}, Chun-Wei LIN^b, Tzung-Pei HONG^c, Vinh V. VU^d, Minh NGUYEN^e and Bac LE^f

^a*Information Technology College, Ho Chi Minh City, Viet Nam*

^b*Innovative Information Industry Research Center, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China*

^c*Department of CSIE, National University of Kaohsiung, Taiwan, R.O.C*

^d*University of Food Industry, Ho Chi Minh City, Viet Nam*

^e*Institute of Information Technology, Ho Chi Minh City, Viet Nam*

^f*University of Science, Ho Chi Minh City, Viet Nam*

Abstract. This paper proposes an algorithm to hide quickly high utility sensitive itemsets. To reduce the execution time, when an item from the sensitive itemset is selected, the rated value is calculated from each transaction containing this item and all values which need be hidden are modified at the same time. Experimental results show that the proposed algorithm improves significantly the runtime.

Keywords. data mining, high utility itemset, privacy preserving, utility.

1. Introduction

Mining high utility itemsets is to find all the itemsets which their utility values are greater than or equal to a user specified threshold. In 2004, Yao et al. [1] proposed the approach of HUI mining from database. Recently, some algorithms of mining HUIs based on transaction weighted utility (twu) are proposed. Firstly, Liu, Liao and Choudhary [2] presented the Two-phase algorithm, then some more effective algorithms are also proposed: The algorithm based on FP-tree (HUC-Prune [3]), the algorithms based on WIT-tree (TWU-Mining [4]). They are very useful in business but they are also very dangerous when this data is not secure.

In the past decade, Saygin et al. [5] discovered the risks of data mining and introduced the solutions for preserving privacy. Evfimievski et al. [6] gave out a good example of necessary of protecting privacy. The clients expect the server to collect all statistic information about the association items from clients' data to provide requests to their customers. However, the clients do not want the server get itemsets which are high sensitive. Thus, if the clients send data to the server, some sensitive itemsets will be hidden by preserving privacy policy.

Although many algorithms of mining HUIs have been developed in recent years, there are few researches in privacy-preserving utility mining (PPUM). In 2010, Yeh

¹ Corresponding Author: Bay Vo, Information Technology College, 12 Trinh Dinh Thao, Tan Phu, Ho Chi Minh, Viet Nam. Email: vdbay@itec.edu.vn

and Hsu [7] presented two algorithms, HHUIF and MSICF, to achieve the goal of hiding sensitive itemsets so that the adversaries could not mine them from the sanitized database. However, both algorithms allowed modifying the quantitative value of an item which result in the difference between the original database and the sanitized database would be great. Besides, these algorithms take a lot of time to find and hide the highest utility item.

In this paper, we propose an approach for fast hiding sensitive itemsets. To reduce the execution time, when an item from the sensitive itemset is selected, the rated value is calculated from each transaction containing this item and all values which need be hidden are modified at the same time. Our algorithm significantly reduces the time of hiding data.

2. Related Work

2.1. Hiding Sensitive Itemsets

Many efficient data mining techniques have proposed, thus causing the security and privacy issues of confidential information. The privacy preserving data mining (PPDM) techniques has thus become a critical research topic for hiding the secured information. Atallah et al. [8] proposed the protection algorithm for data sanitization to avoid the inference of association rules. Both of the addition and deletion procedures are then used to modify databases for hiding sensitive information. Dasseni et al. [9] proposed a hiding algorithm to reduce the confidence or support values of association rules. It consisted of three heuristic approaches to respectively increase the supports of antecedent parts, to decrease the supports of consequent parts, and to decrease the support of either the antecedent or the consequent parts. In his approach, the sensitive association rules could thus be hidden when the supports or the confidences of sensitive association rules were below minimum support threshold. Oliveira and Zaïane [10] introduced the multiple-rule hiding approach to efficiently hide sensitive itemsets. The database is required to scan twice for hiding the sensitive itemsets. In the first database scan, the index file was produced to efficiently find sensitive itemsets within transactions. Three algorithms were then used in the second database scan to remove minimal individual items. Amiri [11] also proposed three heuristic algorithms to hide multiple sensitive rules. Pontikakis et al. [12] then proposed two data distortion approaches to hide the sensitive itemsets. Hong et al. then proposed three approaches to partially delete the items within the transactions or the whole transactions from the original database for hiding sensitive itemsets [13, 14, 15]. In addition to remove the items or entire transactions from the original database, Hong et al. [16] also proposed an approach to add the dummy transactions into the original database by reducing the supports for hiding sensitive itemsets.

2.2. HHUIF Algorithm [7]

The main purpose of HHUIF algorithm is to decrease the utility value of sensitive itemset by modifying the quantity value of items which are contained in the sensitive itemset. In order to decrease the utility value of a sensitive itemset S_i , HHUIF

determines $diff = u(S_i) - \text{minutil}$, $diff$ is utility that need to be decreased of S_i . Then, HHUIF select item $i_p \in S_i$ and transaction $T_q \supseteq S$ such that the utility value of i_p in T_q is maximal for a given sensitive itemset S_i and modify that value. If the utility of i_p on T_q is less than $diff$, it would reduce the quantity of item i_p in T_q to 0. On the contrary, the

quantity of item i_p in T_q would be reduced as $\left\lceil \frac{diff}{s(i_p)} \right\rceil$, where $s(i_p)$ is the profit of item i_p . $Diff$ is updated by using the following formula:

$$diff = \begin{cases} diff - u(i_p, T_q) & \text{if } u(i_p, T_p) < diff \\ 0 & \text{if } u(i_p, T_p) > diff \end{cases}.$$

If $diff$ is still greater than 0, HHUIF continue to repeat all above steps until $diff \leq 0$ and the process of decreasing the utility value of S_i is finished. HHUIF performs the process for all sensitive itemsets until the utility values of each sensitive itemsets are below the minimum utility threshold.

3. Proposed Algorithm

To avoid HHUIF's limitations that presented in above section, we propose an algorithm for hiding sensitive itemsets. The proposed algorithm considers each sensitive itemset S_i in order. For each S_i , it calculates the difference $diff = u(S_i) - \text{minutil}$. If $diff > 0$, we in turn calculate the total value of each item in S_i and select the item with the highest overall sensitivity. Because an item is present in many transactions with different quantity, we calculate $\alpha = o(i_p, T_p) \times s(i_p) / \text{sum}(i_p)$ where α is the rate of decrease on quantity in each transaction T_p . After reduction is completed, we update $diff$. If $diff$ is still greater than zero we continue to find the other i_p which have highest overall value and reduce until $diff \leq 0$. If $diff \leq 0$, the process is stop.

In fact, the rates α of item i_p in the transactions are the same. So, we only need to calculate this value before the loop for each in line 6. Besides, to avoid modifying the transactions of item to zero, the algorithm would assign the value of $o(i_p, T_q) = 1$ if $o(i_p, T_q) \leq 0$, except it is 1 before modifying.

In addition, to ensure the ability of the reducing quantity is minimum, for the last transaction of item i_p , the algorithm will subtract the total reduced values calculate by previous transactions from total values instead of calculating by percentage ratio. This has done by using a variable to maintain the number of reduced values of the previous transactions.

The proposed algorithm:

Input: the original database: DB, the minimum utility threshold $minutil$, the sensitive itemsets U

Output: the sanitized database DB' so that U cannot be mined

1. **for each** sensitive itemset $S_i \in U$
2. $diff = u(S_i) - minutil$ // the sensitivity for modifying
3. Calculator $sum(i_p) = \sum_{T_k \subseteq T(S_i)} u(i_p, T_k)$ for all $i_p \in S_i$
4. **while** ($diff > 0$) {
5. Select i_p such that $sum(i_p)$ is maximum
6. for each T_q where $i_p \in T_q$: $\alpha = u(i_p)/sum(i_p)$
7. Modify $o(i_p, T_q)$ by $\begin{cases} o(i_p, T_q) - o(i_p, T_q) * \alpha & \text{if } \alpha < 1 \\ 1 & \text{if } \alpha \geq 1 \end{cases}$
8. Modify $diff$ and $sum(i_p)$ according to $o(i_p, T_q)$
9. **return** the sanitized database DB'

(where $s(i_p)$ is the profit of item i_p)

Figure 1. The proposed algorithm

4. An Example

Consider the sensitive itemsets which are mined from database in Tables 1 and 2 with $minutil = 120$, we have $U = \{BED, BD, BE, B\}$ as in Table 3.

Table 1. An example transaction database					
Item	A	B	C	D	E
TID					
T1	0	0	16	0	1
T2	0	12	0	2	1
T3	2	0	1	0	1
T4	1	0	0	2	1
T5	0	0	4	0	2
T6	1	2	0	0	0
T7	0	20	0	2	1
T8	3	0	25	6	1
T9	1	2	0	0	0
T10	0	12	2	0	2

Table 2. An example utility table

Item	Profit
A	3
B	5
C	1
D	3
E	5

Table 3. The sensitive itemsets

HUI	Profit
BED	182
BD	172
BE	240
B	240

Firstly, consider $S_i = BED$ with $u = 182 \Rightarrow diff = 182 - 120 = 62$, BED appear in transactions T_2 and T_7 . We have results showed in Table 4.

Table 4. The transactions containing BED and total utility of each item

Item TID	B	D	E
T_2	12	2	1
T_7	20	2	1
Sum u	160	12	10

Thus, item B has the greatest utility. We proceed to modify the quantities of item B in transactions T_2 and T_7 . Before modifying, we calculate the quantity need to be reduced as $\lceil 62/5 \rceil = 13$, from that, we have the decrease ratio $\alpha = 13 \times 5/160 = 40.6\%$. So the quantity of reduced value of item B in T_2 is $40.6 \times 12/100 = 5$, and in T_7 is $13 - 5 = 8$.

After modifying, the result has shown in Table 5.

Table 5. Result after modifying the values of the transactions containing item B

Item TID	B	D	E
T_2	7	2	1
T_7	12	2	1

Table 6. HUI and their utilities values after modifying the transactions containing item B

HUI	Utility Value
BED	117
BD	107
BE	175
B	175

Continue to select $S_i = BE$ and do the same steps. Item B is selected because B has the highest total values of utility. Then the $diff = 175 - 120 = 55$. Then, we calculate the value of $\alpha = 55 / 155 = 35.5\%$ and the quantity should be reduced is $55 / 5 = 11$. Thus item B in transaction T_2 should be reduced $\lceil 7 \times 35.5/100 \rceil = 3$. Item B in transaction T_7 should be reduced $\lceil 12 \times 35.5/100 \rceil = 5 \Rightarrow$ Item B in transaction T_{10} should be reduced $11 - (5+3) = 3$.

5. Experimental Results

Experimental results are performed on a Centrino Core 2 Duo (2×2.53 GHz) PC with 4 GBs of RAM and running in Windows 7. The algorithms were implemented in C# (2008). The experimental database (BMS-POS) was downloaded from <http://fimi.cs.helsinki.fi/data/> having characteristics as in Table 7.

Table 7. Experimental databases downloaded from <http://fimi.cs.helsinki.fi/data/>

Database	#Trans	#Item	Note
BMS-POS	515597	1656	Modified

We modified this database by adding value column (random in range of 1 to 10) for each item corresponding to each transaction, and create one more table to store benefit values of items (value in range of 1 to 10).

To easily observe the results in time comparison, we present the results of the time for hiding all sensitive HUI from the above database.

Table 8. The number of HUIs corresponds to the database in Table 7

Database	minutil (%)	#HUIs
BMS-POS	5	4
	4	6
	3	7
	2	22

From Table 8, we can see that the number of HUIs in databases is often not large. The largest quantity in above experiments just is 22.

The following section presents the results in comparison of execution time between HHUIF algorithm [7] and the proposed algorithm.

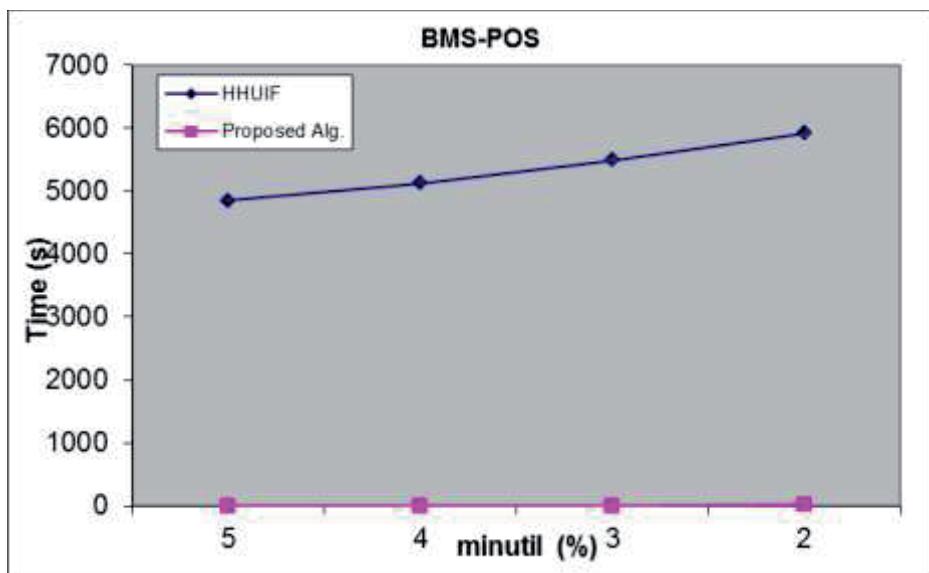


Figure 2. Performance comparison between HHUIF algorithm and the proposed algorithm in BMS-POS database

Figure 2 shows the performance comparison about execution time between HHUIF algorithm and the proposed algorithm in BMS-POS database. It is possible to find that the time difference of two algorithms is quite large. For example, with = 5%, the runtime of HHUIF is 4849 seconds, meanwhile the runtime of the proposed algorithm is 2 seconds. The result is also different when considering remain minutis.

6. Conclusions and Future Work

This paper has proposed a new algorithm that outperforms HHUIF both the runtime and the difference between sanitized database and original database. For the runtime, the proposed algorithm is much faster than HHUIF algorithm because for each selected item, it modifies multiple transactions at once while HHUIF only modifies one after one.

As we know, time is just a factor in the problem of hiding high utility sensitive itemsets. The problem is that how to hide the sensitive itemsets with some requests, such as: 1) Difficult to recover values being modified; 2) The gap between the original database and the sanitized database is as minimum as possible. Therefore, in future, we are going to do research in these problems aiming to develop efficient algorithms for hiding high utility sensitive itemsets.

References

- [1] H. Yao, H.J. Hamilton, C. J., Butz, *A foundational approach to mining Itemset Utilities from Databases*, Proceedings SIAM International Conference on Data Mining, 2004, 482 – 486.
- [2] Y. Liu, W. Liao, A. Choudhary, *A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets*. In PAKDD 2005, Hanoi, Viet Nam, 2005, 689-695.
- [3] C. F. Ahmed, S.K. Tanbeer, B.S. Jeong, and Y.K., Lee. *An Efficient Candidate Pruning Technique for High Utility Pattern Mining*. In PAKDD 2009, LNAI 5476, 2009, 749–756.
- [4] B. Le, H. Nguyen, and B. Vo. *An efficient strategy for mining high utility itemsets*. In International Journal of Intelligent Information and Database Systems 5(2) (2011), 164-176.
- [5] Y. Saygin, V.S Verykios, C. Clifton, *Using Unknowns to Prevent Discovery of Association Rules*. SIGMOD Record 30(4) 2001, 45-54
- [6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. *Privacy preserving mining of association rules*. In Information Systems 29(4) (2004), 343-364.
- [7] S. Yeh, P. C. Hsu. *HHUIF and MSICF: Novel algorithms for privacy preserving utility mining*. Expert Systems with Applications 37(7) (2010), 4779–4786.
- [8] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios. *Disclosure limitation of sensitive rules*. In Knowledge and Data Engineering Exchange Workshop, 1999, 45-52.
- [9] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. *Hiding association rules by using confidence and support*. In Proceedings of the 4th international workshop on information hiding, Springer, 2001, 369 – 383.
- [10] S. R. M. Oliveira and O. R. Zaïane. *Privacy preserving frequent itemset mining*. In IEEE International Conference on Privacy, Security and Data Mining, 2002, 43-54.
- [11] A. Amiri. *Dare to share: Protecting sensitive knowledge with data sanitization*. In Decision Support Systems 43(1) (2007), 181-191.
- [12] E. D. Pontikakis, A. A. Tsitsonis, V. S. Verykios. *An experimental study of distortion-based techniques for association rule hiding*. In IFIP Workshop on Database Security, 2004, 325-339.
- [13] T. P. Hong, K. T. Yang, C. W. Lin, and S. L. Wang. *Evolutionary privacy-preserving data mining*. In World Automation Congress, 2010, 1-7.
- [14] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang. *A heuristic data-sanitization approach based on tf-idf*. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2011, 156-164.

- [15] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang. *A lattice-based data sanitization approach*. In IEEE International Conference on Systems, Man, and Cybernetics, 2011, 2325-2329.
- [16] T. P. Hong, C. W. Lin, C. C. Chang, and S. L. Wang. *Hiding sensitive itemsets by inserting dummy transactions*. In IEEE International Conference on Granular Computing, 2011, 246-249.

A Method for the Generation of Semantic Annotation from Sport News Using Ontology Based Patterns

Quang-Minh NGUYEN^{a,1}, Tuan-Dung CAO^b, Thanh-Hien PHAN^b,

Hoang-Cong NGUYEN^b and Tatsuya HAGINO^c

^a*School of Electronics and Telecommunications,*

Hanoi University of Science and Technology, Vietnam

^b*School of Information and Communication Technology,*

Hanoi University of Science and Technology, Vietnam

^c*Faculty of Environmental Information, Keio University, Japan*

Abstract. In the framework of this research, we focus on describing the algorithm to collect, process information in natural language to turn it into semantic annotations based on ontology, patterns and the knowledge ground in sport. This paper will concentrate to solve the problem of enhancing the named entities recognition by developing ontology and enriching the knowledge base for the recognition engine. Besides this, it is at the same time solving the problem of semantic recognition in texts using ontology-based patterns, harnessing the power of ontology in identifying synonyms, aliases.

Keywords. Semantic web, sport ontology, information extraction, semantic annotation

1. Introduction

Sport is well-known as an interesting area that attracts hundreds of millions of people around the world. When a sport event is happened, there are many sources of supplied available information, included published discussions revolving statements around the event by many different aspects. From this, it shows necessity of building an information system capable of collecting, integrating and extracting the information automatically (or semi-automatically) from a variety of sources. This is to ensure that readers could approach multidimensional information before events, certain sport character occurring around the world; and also serving other important purposes such as searching, statistics, connections, etc.

At present, many research projects have been launched to address the problem under a variety of different approaches. The general idea is to build a common information performance model, so that information from different sources can be performed in the same standard, semantics. By this performance, it is easy to be

¹ Corresponding Author: Quang-Minh Nguyen, Hanoi University of Science and Technology, 1 Dai Co Viet Road, Hanoi, Vietnam; E-mail: minh.nguyenquang@hust.vn.

reciprocally transformed between information formats to support the link, search and combination of information for many different purposes.

One of approaches that give the most promising results is to use a semantic web technology. The researches have showed the advantages created by the semantic web technology, for example, the ability to link data from different sources in many different areas such as the BBC system [1], the ability to integrate information from heterogeneous data sources in a certain field such as INDUS (Intelligent Data Understanding System) system in biology [2], or the ability to search for semantics, answering questions in the research of Abir Qasem and colleagues [3]. In addition, with the semantic web, systems have the ability of deduction to self-generate new knowledge from existing knowledge etc. In the line with this trend, we have also used semantic web as a technology platform for our research. In a recent work [4], we have proposed an integrated sport information system of which ontology plays a central and core role. Following by this trend, there is a major challenge to be encountered related to natural language processing for creating quality semantic annotations from traditional text data.

The researches of the generation of semantic annotation from texts have been interested by scientists. In addition, many different technical platforms such as pattern-based [5], neural-networks, structure-analysis, supervised learning [6], ontology-based [7], etc. have been studied. Each of them has its own advantages and disadvantages but most are still simple in order to be applied for general texts, and do not focus specifically on a certain field.

This article focuses on describing the algorithm of semantic extraction from texts in sport of flexible combination between ontology-based and pattern-based methods. It is concentrating of developing ontology and enriching the knowledge base of sports in order to increase named entities recognition as well as to recognize the semantics of texts with ontology-based patterns. We use the power of ontology to recognize semantics even in the case of texts using synonyms, abbreviations or aliases.

Section 3 will present the approach and building of BkSport ontology for sport sector and the roles of this ontology in approaching. Section 4 indicates the process of formation and development of the knowledge base in sport. The most important part of this article is in section 5, where the algorithm of detection and generation of semantic annotation will be described in detail. Section 6 and section 7 present the experiments, conclusion and future works.

2. Related Work

The semantic recognition, extraction from texts is an important issue and plays a decisive role in the effectiveness of the use of semantic web technology in the systems of collecting, integrating knowledge from various sources. Therefore, many research groups around the world have participated in solving this problem by many different approaches. C-Pankow (Context-driven PANKOW) [5] is the improvement of Pankow (Pattern-based Annotation through Knowledge on the Web). This is a study on the entities recognition and the generation of semantic annotation for such entities in an automatic or semi-automatic way on websites. C-PANKOW builds the library of its patterns to recognize the entities on the web. With each captured entity, C-PANKOW conducts queries on the Google, retrieves abstracts and picks out the appropriate

abstracts based on the degree of similarity of such abstracts with the content of website and thereby automatically generating semantic annotations on captured instances. MEAT [7] shows the approach using the semantic web technology for the generation and use of ontology-based semantic annotations. The system also uses GATE and JAPE rules to recognize relations and terms in the biological domain then generate semantic annotations on these terms. Ontea [8], a pattern based semi-automatic Text Mining and Information Retrieval, gives the solution that is the combination between traditional information retrieval methods and the method of using semantic web to increase the relevancy of automatic annotation on the web. It processes texts or documents of an application domain that are described by a domain ontological model and uses regular expressions to identify relations between text and a semantic model.

3. Design and Building of an Ontology for the Sport Domain

Like other semantic applications, our algorithm is conducted based on ontology in sport named BKSport. Generally, the building of ontology plays a very significant role. The details of the concept and its properties must be accurate and consistent with sufficient details. If the tree of the concept and details is too specific, it will be possible to easily create ambiguity and redundant data. In contrast, if the ontology only stops at high level, it will not be strong enough to represent knowledge in a diverse and abundant way. In the process of ontology development, we have referred to the ontology that is now being used in the sports field of BBC [9] and the ontology serving for the semantic annotation- PROTON, thereby evaluating the suitability for re-use of some existing concepts and properties and expansion, definition of new concepts and properties.

BBC - Sport Ontology was originally built by for a specific use case of the BBC, mainly in consistency with the structure of an Olympic tournament and at the simple level. However it can be widely used for events, sports competitions in general. This is a simple ontology that allows sports data to be represented with semantics in such aspects as: The structure of a sport competition as a sequence of events, the internal competition of a tournament, the rules of a tournament, the connection of a prize with a tournament and how to achieve it, etc.

Our method uses KIM's [10] API, and thus during the design and building, our research team has also studied the PROTON ontology that KIM used. PROTON was developed in the framework of the project SEKT (<http://www.sekt-project.com/>), defining about 250 concepts and 100 properties providing most of the high-level concepts needed for semantic annotation, indexing and retrieval.

3.1. The BkSport Ontology

The BkSport ontology was built with high coverage of the sports of BBC - Sport Ontology, and simultaneously particularized in each specific subject to allow the recognition of specific relationships (e.g. "transfer" relations in football). Ontology also ensures the compatibility with PROTON, which is shown in the concept mapping capabilities. The figure below shows an example of the concept mapping from BkSport to the PROTON (see Figure 1):

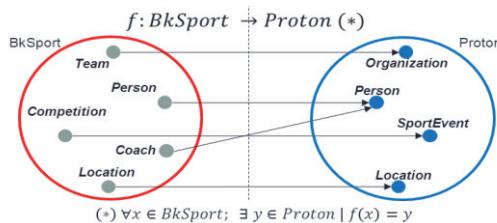


Figure 1. Mapping from BkSport to PROTON.

In order to achieve the best results, the research team focused on developing the concepts in sport in general and going deeply into the two sports of the most interest including football and tennis.

Most sports newsletters focus on some of the main concerns of readers such as "Event" (e.g. tournament, game, etc.), "Person" (e.g. football player, tennis player, coach, etc.) or prestigious awards, so BkSport has also focused on developing concepts, relations mainly revolving around these concepts. Figure 2 shows the key concepts and relationships in BkSport. It is easy to find out that the relationship is mainly focused on some important concepts such as Competition, Person, Team, etc. Besides the building of the concepts, the building of the properties representing the relationships among concepts is also great importance. These properties are the basis for building the recognition patterns and generating semantic annotations afterwards. The research team focused on developing the attributes to describe the relationships of regular concern in sport such as the information results of a match (hasResult, win, etc.), the statements of organizations, individuals (saidThat). Apart from that, attributes that describe the relationships revolving around the subjects of public interest have been also built such as information on players, awards (hasAward), etc.

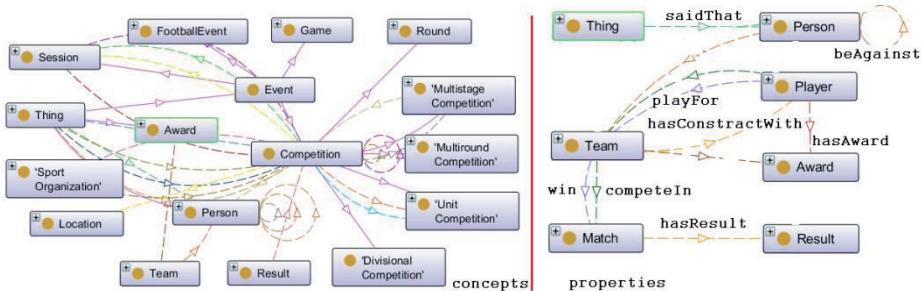


Figure 2. Diagram of main concepts in The BkSport ontology.

4. Building and Enrichment of Knowledge Base in Sport

The ability to detect entities and recognize semantics in texts of the system depends on knowledge base with sufficient coverage in sport, such as players (name, age, sport subject, etc.), clubs (name, home yard, etc.), competitions, awards, stadiums, etc. Some well-known and prestigious websites in sport such as espn.go.com, skysport.com, atpworldtour.com, etc. have been selected as data sources. Figure 3 shows the process of enriching the knowledge base in a semi-automatic way.

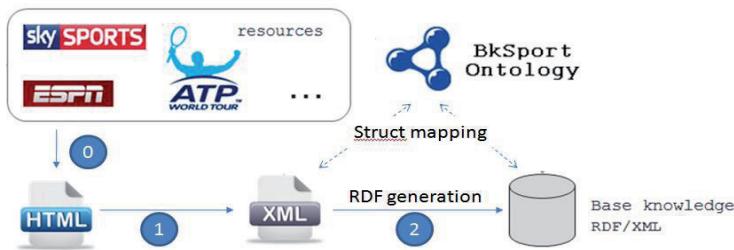


Figure 3. Building of knowledge base.

In step 0, from reputable sources, the system captures links to profiles of players, clubs, etc. All webpages of these links then have been downloaded for the system to process, extract information, and build knowledge base. The profiles are often structurally performed, thereby in step 1 by analyzing the structure of webpage's source code, the system retrieves the necessary information and performs them in the form of XML (Attributive cards are designed to be able to map on the properties in the ontology). In step 2, the reference set based on the suitability of the content, meanings between the XML tags and attributes, concepts in ontology are used to convert information from XML format to RDF/XML format. For example, the information of tennis player Novak Djokovic, after preliminary treatment and implementation of step 1 and 2 will be expressed as follow (see Figure 4):

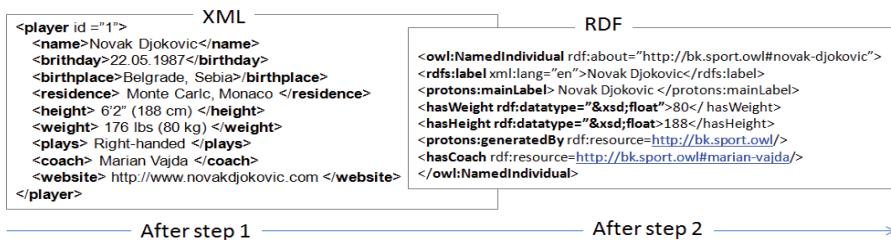


Figure 4. Data after transformation to RDF/XML representation.

We used XSLT to transform XML data into the knowledge base (described as using RDF / OWL). Currently, the knowledge base focuses on both football and tennis. This data will be enriched over time and expanded to other sports.

5. Named Entities Recognition and Generation of Semantic Annotation

The knowledge base, a set of recognition rules, the BkSport ontology and a set of articles will be the input of the recognition algorithm. The ultimate goal is to detect the entities (of the concepts defined in BkSport) as well as the relationships between these entities, thereby extracting the corresponding semantic annotations. The overall processing is described in Figure 5:

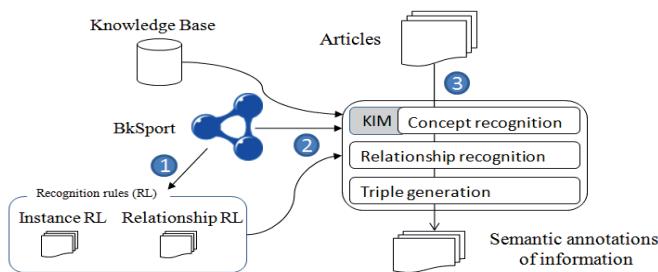


Figure 5. Named entities recognition and generation of semantic annotation.

The first and foremost step is to help improve the ability of named entities recognition as well as the relationships recognition. In this step, we rely on the BkSport ontology and the structural analysis of natural language (English) to build a set of named entities recognition rules and recognize the relationships that appear in the texts. In step 2, the BkSport ontology will be mapped into PROTON, at the same time knowledge base will be loaded to support KIM to detect entities. Step 3 is to analyze the texts to define entities, relationships, thereby extracting semantic data about the articles. The details of these issues will be clarified as follows:

5.1. Named Entities Recognition with KIM-API

KIM's [10] operating is based on PROTON ontology. In order to recognize entities at the detailed level in sport using this platform, the mapping of the concepts from BkSport into PROTON is indispensable. This process was conducted from establishing on mapping function mentioned in the introduction part.

The nature of this process is to use subClassOf property (subclass of a class) to attach the concepts of BkSport into PROTON (protons, protont and protonu). We call $f:BkSport \rightarrow PROTON$ and C_1, C_2 are concepts which are defined in BkSport ontology and PROTON($C_1 \in BkSport, C_2 \in PROTON$). If $f(C_1) = C_2$, the following statement is generated:

```
bksport:C1 rdfs:subClassOf proton:C2
```

For example: The concept of Competition is mapped to the concept of SportEvent in the proton ontology, then we have the statement:

```
bksport:Competition rdfs:subClassOf protonu:sportEvent
```

In addition, we have set up Gazetteer (resources in KIM that use a dictionary to annotate text) to collect named entities in the BkSport knowledge base by writing semantic queries in KIM to get the labels, URI, and the classes of such named entities. All named entities extracted from the KIM API are represented in a specific KIM structure which called KIMAnnotation, including the information on the location in the article, the name of captured entities, entity URI, etc.

5.2. Improvements of Named Entities Recognition with BkSport Ontology

Since the detailed levels of the PROTON ontology and the BkSport ontology are different (BkSport represents the concepts at a deeper and more detailed level in the

field of sports), the mapping function f is not the single mapping (e.g. "Striker" or "Defender" is understood as a Person by KIM). To solve this problem, we have designed patterns using JAPE rules to define the recognition rules based on the BkSport ontology and the obtained KIMAnnotation. Improvements have been realized on the following cases:

Recognition of entities at the detailed conceptual level: The goal is detecting the entities of detailed classes in the BkSport ontology such as "Defender" or "Forward" instead of high-level concepts such as "Person" or "Player". The analysis has showed that most of entities are represented as "occupation" + "private name" (e.g. Goalkeeper van der Sar, Striker Messi, etc.). Occupations are often labels of concepts, therefore; the algorithm uses labels of concepts to be patterns to build up recognition rules for entities of each concept. For example, the rule for detecting a "Defender" is defined as follows: If the term "Defender" or "defender" is followed by a named entity that is recognized as "Person", such entity should be a "Defender".

Entity recognition written in a shortened name: In texts, after the full name of the entity is used, it will be repeated with a shortened name to make the article short and easy to read (e.g. "Lionel Messi" is written into "Messi"). Therefore, it is important for the improvement to provide the ability to recognize the entity when its name is shortened. The shortened name will usually be part of the full name, hence; an entity with the shortened name can be recognized when it is detected with the previous full name by using operations of partly matching.

Recognition of entities of the same name but different type: This is also a common case that is found in semantic annotation from texts when there are two entities with the same name appearing in the texts. For example, Giuseppe Meazza is the name of a player, but also the name of a stadium. In this case, the solution is also similar to the first case, we will carry out sample check, identify the type of entity depending on the following suffixes. In above example, if Giuseppe Meazza is followed by the concept of a stadium, the captured entity will be the stadium.

5.3. Generation of Semantic Annotation

To generate semantic annotation of a text, in addition to the recognition of named entities, it is very important to detect relationships appearing in the text. Table 1 shows the keywords appearing in common relationships in the sport. We focus on three main groups of relationships that are mostly interested by audiences: relationships between person and person (<Person><relationship><Person>), relationships between person and organization (<Person><relationship><Organization>) and relationships between organization and organization (<Organization><relationship><Organization>). Each relationship associates with a set of specific words when appearing in the text. This set of keywords serves as a basis for building relationship recognition patterns. As mentioned above in section 3.1, relationships of regular concern in sport are defined during the process of ontology development. Corresponding keywords were selected manually by human experts in Table 1.

Table 1.The keywords appearing in common relationships in the sport

Relationship	Keys	Pattern
defeat	"win", "beat", "defeat", "overpower"	{SportTeam} [key] {SportTeam}

faceWith	“face with”, “against”	{SportPerson} [key] {SportPerson}
leftOnBenchFor	“left on bench for”	{SportPerson} [key] {SportCompetition}
hasRetired	“retires”, “retired”	{SportPerson} [key]
praise	“praise”, “praises”	{SportPerson} [key] {SportPerson}

Apart from normal relationships, indirect statements are very often given in the sport news. Like other relationships, this relationship is also recognized based on the patterns built from a set of keywords describing the relationships, Table 2 describes keywords and patterns to recognize this relationship:

Table 2. The keywords for indirect statements

Keywords	Pattern
“say that”, “said that”, “announce”, “speech”	{SportPerson} [key] {Statement}
“statement”, “add”, “added”	{Statement}, {SportPerson} [key]

From relationships described in the BkSport ontology and their respective patterns, we have used JAPE to build the relation recognition rules. All of them operate in accordance with a general principle: If a pattern is found, the corresponding relationship will be generated.

Particularly for the case of *indirect speech recognition*, we emphasize to analysis of followed indirect clauses (followed by "said that", "announce"). The recognition and generation of semantic annotations in this case are conducted as follows:

```
// P is a reification pattern (e.g. A "said that" B, A "announce" B, etc.)
P = {A "said That"/"announce" B};
foreach(Annotation p in P) do{
    statement = p.get("B");
    // annotates statement
    annotationSet = BKSport.annotate(statement);
    for each(Annotation annotation in annotationSet){
        if(annotation.contains("semantic")){
            // Creates statement which is same annotation
            subject= annotation.get("subject");
            predicate= annotation.get("predicate");
            object= annotation.get("object");
            // Generate triples:
            <A><bksport:saidThat><statement>;
            <statement><rdf:subject> subject;
            <statement><rdf:predicate> predicate;
            <statement><rdf:object> object;
            }
        }
    }
}
```

6. Experimentation

The experiment has been performed on a set of 130 sport news in football and tennis. News was downloaded manually and randomly from resources including skysports.com, premierleague.com, guardian.co.uk, espn.go.com, bbc.co.uk/sport and etc.

Figure 6 shows recognition results applied on sport news “Cisse hits two as Magpies sink Baggies”. Figure 7 illustrates indirect speech recognition by the system. We evaluate the algorithm in two main tasks:

- Detect and generate instances of ontology in sport domain
- Detect certain semantics of news to generate additional semantic annotation

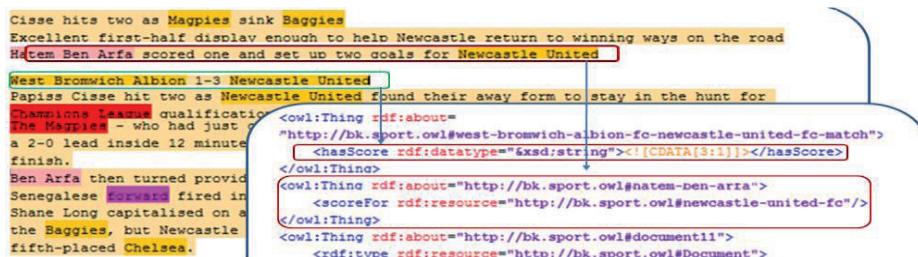


Figure 6. Semantic information extracted by BkSport.

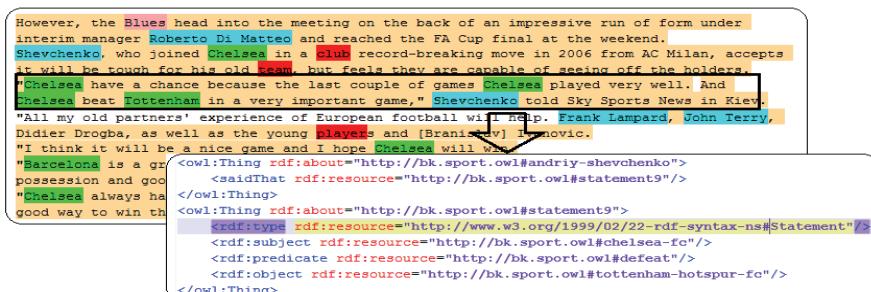


Figure 7. Reification extracted by BkSport.

In order to evaluate the performance of the algorithm, the standard recall (1) and precision (2) are applied. Recall (1) is defined as the ratio of correct positive predictions what are made by the system and the total number of positive examples. Precision (2) is defined as the ratio of correct positive predictions what is made by system and total number of positive predictions:

$$R = \frac{\text{Relevant recognized instances (triples)}(\text{RR})}{\text{Total relevant instances (triples)}(\text{TRE})} \cdot 100 (\%) \quad (1)$$

$$P = \frac{\text{Relevant recognized instances (triples)}(\text{RR})}{\text{Total recognized instances (triples)}(\text{TR})} \cdot 100 (\%) \quad (2)$$

To obtain these values, the algorithm was tested on each of news, and semantic annotations were created manually at the same time. Table 3 shows that the improved algorithm gets better precision and recall value in comparison to what of previous research [4] both in the named entities detection and semantic annotation generation. This can be explained by two reasons. Firstly, the algorithm exploits ontology and knowledge base to eliminate ambiguity caused by homonym terms in sport domain. Secondly, the dataset are built up purely from sport news, not on the mixture of multi domains. In addition, complicated triples such as reification are now detected and generated. This is resulted from applying of patterns and rules proposed in the section 5. However, the recall is still need to be improved because the volume of the data in the knowledge base of BkSport has not been large and abundant enough. Besides, the number of patterns to recognize relationships is not adequate to cover all cases.

Table 3. Named entities and triples recognition statistic

	TR	RR	TRE	P (%)	R (%)
Named entities	2699	2692	4415	99,74	60,97
Triples	1002	890	1663	88,82	53,52

7. Conclusions and Future Work

The semantic annotation from texts is a research problem with many different approaches. This article has presented the usage of ontology-based pattern to recognize entities and to detect the knowledge in sport, a domain that still lacks the similar research. Improvements in the algorithm have brought positive and promising results. Our research objective is to extract simple semantics from text that are sometimes important (e.g. from title, speech or short sentences). Recognizing all complex semantics is not a practical approach. The approach using ontology-based patterns has proved to be consistent with this objective. Moreover, this method can also be applied in other domains if appropriate ontologies and patterns are adaptively built.

In coming time, we will focus on enriching the knowledge base to improve the named entities recognition as well as discover new patterns to increase the relationship detection between instances. We are also interested in comparing the similarity of short sentences in the news with sentence patterns exploiting the available semantics [11]. In addition, we will implement our approach to build a semantic portal of sport data in multi-agents paradigm. In this society of agents, each group of agents is dedicated to a specific role such as: monitor and collect data from information sources, update the knowledge base of sports, generate semantic annotation and carry out the semantic search and matching information.

References

- [1] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, Robert Lee, Media meets semantic web – How the BBC uses DBpedia and Linked Data to make connections, *ESWC 2009 Heraklion Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications* (2009), 723-737.
- [2] Doina Caragea, Jie Bao, Jyotishman Pathak, Adrian Silvescu, Carson Andorf, Drena Dobbs, Vasant Honavar, Information Integration from Semantically Heterogeneous Biological Data Sources, *Proceedings of the 16th International Workshop on Database and Expert Systems Applications* (2005), 580-584.
- [3] Abir Qasem, Dimitre A. Dimitrov, Jeff Heflin, Efficient Selection and Integration of Data Sources for Answering Semantic Web Queries, *Proceedings of the Second IEEE International Conference on Semantic Computing* (2008), 245-252.
- [4] Quang-Minh Nguyen, Tuan-Dung Cao, Hoang-Cong Nguyen, Tatsuya Hagino, Towards efficient sport data integration through semantic annotation, *The Fourth International Conference on Knowledge and Systems Engineering KSE2012* (2012), 99-106.
- [5] Philipp Cimiano, Günter Ladwig, Steffen Staab, Gimme' The Context: Context-driven Automatic Semantic Annotation with C-PANKOW, *Proceedings of the 14th international conference on World Wide Web WWW2005* (2005), 332-341.
- [6] Andrew Carlson, Justin Betteridge, Richard C. Wang , Estevam R. Hruschka Jr., Tom M. Mitchell, Coupled semi-supervised learning for information extraction, *WSDM '10 Proceedings of the third ACM international conference on Web search and data mining* (2010), 101-110.
- [7] Khaled Khelif, Rose Dieng-Kuntz, Pascal Barbry, An Ontology-based Approach to Support Text Mining and Information Retrieval in the Biological Domain, *Journal of Universal Computer Science* vol. **13 no. 12** (2007), 1881-1907.
- [8] Michal Laclavík, Marek Ciglan, Martin Šeleng, Stanislav Krajčí, Ontea: Semi-automatic Pattern based Text Annotation empowered with Information Retrieval Methods, *Tools for Acquisition, Organisation and Presenting of Information and Knowledge* (2007), 119-129.
- [9] Jem Rayfield, Paul Wilton, Silver Oliver, *Sport ontology*, BBC, United Kingdom, 2011.
- [10] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, Miroslav Goranov, KIM - Semantic Annotation Platform, *Proceedings of Second International Semantic Web Conference* (2003), 834-849.
- [11] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, Keeley Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, *IEEE Transactions on Knowledge and Data Engineering* vol. **18 no. 8** (2006), 1138-1150.

A Multi-Agent Architecture for Health Information Systems

Luca PALAZZO^{a,1}, Matteo ROSSI^a, Aldo Franco DRAGONI^a, Andrea CLAUDI^a,
Gianluca DOLCINI^a and Paolo SERNANI^a

^a*Information Engineering Department, Università Politecnica delle Marche, Italy*

Abstract. The healthcare domain is wide and characterized by system and data heterogeneity. To achieve high quality and efficiency standards, interoperability between different information systems is strongly required.

The aim of this paper is to propose an agent oriented architecture to address this kind of issues, which is compliant with the European Union (EU) guidelines and with requirements issued by the Italian Ministry of Health. To validate and to show the capabilities of our system, we developed on such architecture a typical emergency-response scenario, where a first aid medical staff urgently needs to retrieve, through mobile devices, the Patient Summary (PS) of a citizen, which is part of his Electronic Health Record (EHR).

Keywords. Health Information Systems, Interoperability, Multi-Agent Systems, Patient Summary, EHR, HL7 CDA

1. Introduction

The healthcare domain is facing a growing number of challenges. The incidence of medical errors is rising; many medical facilities are understaffed, and serve increasingly large areas; healthcare costs are rising more and more. The healthcare budgets are shrinking in many countries, and healthcare facilities are under pressure to provide better services with less resources [1]. Health Information Systems (HIS) are at the heart of all these challenges. They can provide a better coordination among medical professionals and facilities, thus reducing the number and incidence of medical errors. In the same time, they can reduce healthcare costs and may provide a means to improve the management of hospitals [2]. Unfortunately, due to the inherent complexity of their application domain, HIS are fragmented in various systems that hardly make use of communication standards, process definition protocols and homogeneous data representations. Much of the research in this field is aimed to address these problems, and different solutions have been proposed during the last years.

¹Corresponding Author: Luca Palazzo, Università Politecnica delle Marche, Dipartimento di Ingegneria dell'Informazione, Via Brecce Bianche, 60131 Ancona, Italy; E-mail: l.palazzo@univpm.it

1.1. Related Works

In recent years, two different technologies have been the subject of much of the research relating to HIS: cloud computing and multi-agent systems. A mobile system that enables electronic healthcare data storage, update and retrieval using Cloud Computing is proposed in [3], in which a mobile application based on an Android client enables the users to retrieve remotely health information and images. In [4] a wireless sensor network is used to automate the data collection process. The collected information are distributed through a Cloud Computing solution to medical staff. An alternative approach is proposed in [5], where data and service interoperability is obtained through a distributed and agent-oriented system. Finally, [6] and [7] use the multi-agent system technology to support the home-care monitoring and treatment of patients.

1.2. Our Contribution

In this paper we propose an agent oriented architecture capable to access geographically distributed data to allow health professionals to retrieve/update any patient's record efficiently and reliably. Such architecture meets the interoperability requirements among different health facilities and, at the same time, integrates with existing legacy systems (including local databases), being a new software layer on top of existing ones: this allows to protect the investments made by facilities and institutions as required by ministerial directives [8], in addition to address interoperability issues.

The main advantages of such architecture are:

- **Distribution.** A key concept of agent technology is flexibility: the complex issues of interoperability and integration with existing systems is broken down to minor tasks assigned to individual agents: cooperation is the solution to the original question. Retrieving data is possible from any point in the territory just through communication of distributed agents, and expensive infrastructures - as happens with Cloud solutions - are not required.
- **High modularity.** Thanks to standardization activities made by the multi-agent systems (MAS) community - FIPA IEEE -, simply adding new agents in the architecture (registering their services and sharing the same ontology) is enough in order to extend the capabilities of the system.
- **Robustness.** An agent oriented infrastructure provides many recovery techniques to better achieve fault tolerance goals.
- **Integration with existing systems.** With the aid of wrapper agents, each one designed for a particular instance of legacy information systems, the architecture represents a higher fully interoperable software layer. Communication at this level is readily able to use well established standard ontologies for messaging (HL7), definition of clinical documents (HL7 CDA), scheduled workflows (IHE) and health care terminologies (such as LOINC and SNOMED CT).

1.3. Paper Structure

The rest of this paper is organized as follow: section 2 details the multi-agent system architecture; section 3 illustrates an implementation related to an emergency-response scenario; section 4 includes a discussion of the results and concludes the paper.

2. Infrastructure

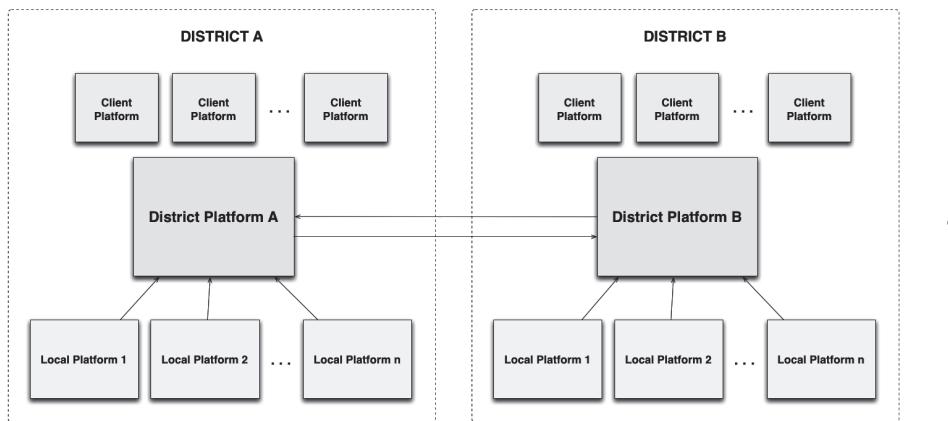


Figure 1. The global architecture is structured in three logic layers.

The agent oriented architecture is expressed by three levels of abstraction, named local platform, district platform and client platform (Fig. 1-2): each one is characterized by its specific agents and resources as described in the next subsections. The discriminating factor between the first two layers is of administrative nature: there is a local platform for each health facilities in the territory (eg. a hospital); facilities refers to administrative districts, which constitute the second layer of the architecture; finally, the client level is represented by any software agent which needs to login to the infrastructure to retrieve documents or insert/update a patient's health record.

2.1. Local Platform

There is a local platform (Fig. 3) for each health facility. It has the role to interface with any information system, currently present in the structure, committed to the management of clinical documents (create, edit, search, access) and the scheduling of different departments in the facility. Every local platform needs to know the address of its referring district platform in order to have access to the entire agent infrastructure.

LocalDBWrapper.

The task of such agents is to interface with the databases of a certain local healthcare institution. The advantages in the use of wrapping agents are the following:

- All the legacy systems would not be modified or replaced, but in fact encapsulated within such agents. In this way, any external agent, which needs to access to data contained by a local database, will be able to obtain them simply by communicating with the referring LocalDBWrapper agent, thus avoiding direct interaction with legacy systems.
- It makes possible to abstract the actual data representation within the different information systems available in the various facilities. With this solution, we don't need to address issues like information conflicts (such as homonymy and syn-

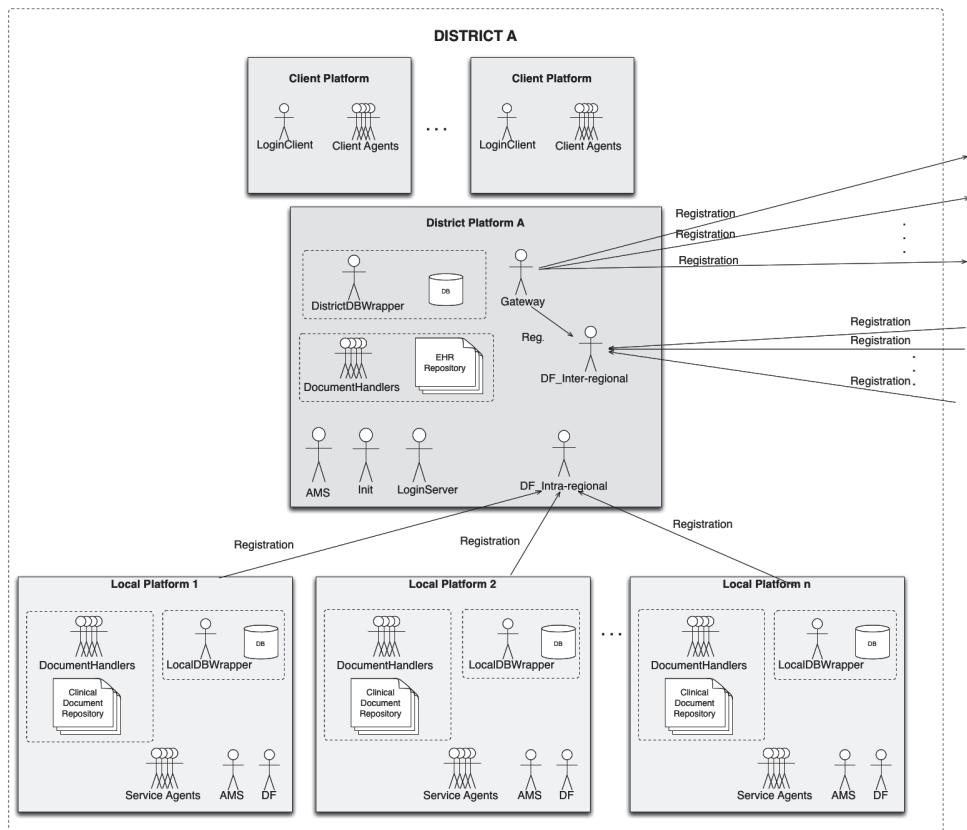


Figure 2. Relation between local platforms and their referring district platform.

onymity) or data schema inconsistencies by burdensome techniques of renaming, restructuring or even system redesign; it is sufficient to design a wrapping agent for each different legacy system able to translate the internal data representation in the ontology shared by all the agents in the infrastructure.

In order to add a local platform to the entire agent oriented architecture, the LocalDB-Wrapper agents must register to DF_Intra-District agent of their referring district platform: this makes it available from distributed and remote agents, which need to retrieve data contained by the local structure.

DocumentHandler.

This kind of agents are able to access the content of a specific clinical document produced within the facility, such as clinical reports, laboratory tests, prescriptions, etc. In general, a DocumentHandler agent is contacted by a client agent to get health records managed by it: the DocumentHandler agent locates the requested document through its unique identifier, obtains it from the clinical repository and translates the information in an outgoing message towards the requesting client agent. Hence, the latter will be able to get the contents of clinical data requested.

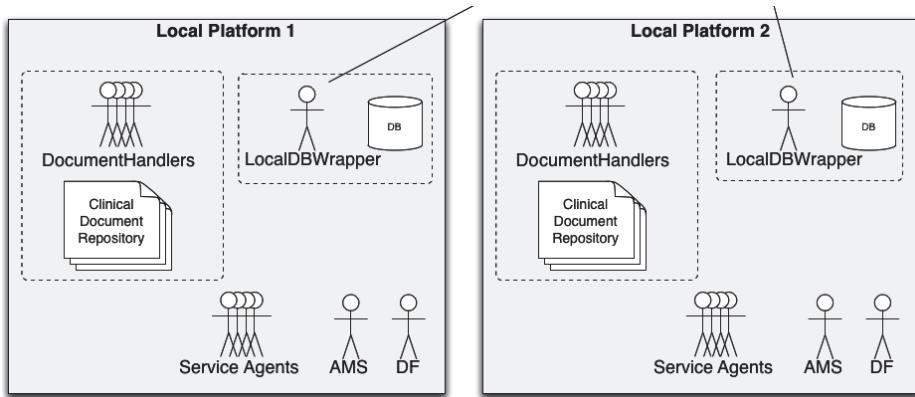


Figure 3. Local platform agents.

Service Agents.

This set consists of agents for the management of different departments of the health-care structure (e.g. radiology, cardiology, analysis laboratory, etc.). This paper does not provide further information on this field, but it is possible to find details about an agent oriented implementation of the Radiology Scheduled Workflow provided by Integrated the Healthcare Enterprise (IHE) consortium in [9].

2.2. District Platform

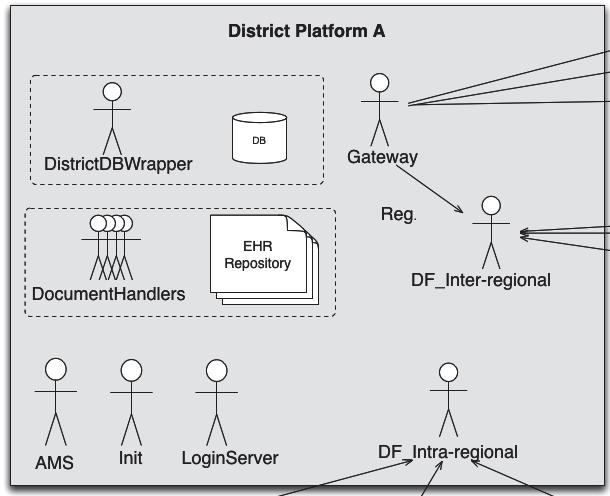


Figure 4. District platform agents.

The main task of a district platform (Fig. 4) is to encapsulate all the local platforms that administratively belong to it. Basically, the district platforms represent the logic layer which composes the final architecture and allows to achieve the interoperability goal of

our distributed system: every district platforms, therefore, must know each other their address.

DistrictDBWrapper.

These agents has similar functions with local wrappers: they manage data within district databases. The gateway agent contacts wrappers in order to store or retrieve any reference to a patient's clinical records, which have been produced by every local platform in the territory or by general practitioners.

DocumentHandler.

DocumentHandler agents manage those kind of documents which are of administrative competence of a district, such as EHR and Patient Summary [10]. They may refer to health records which are distributed in different local platforms: the Gateway agent has the role to look for and gather this information.

Gateway.

The Gateway agent catches the client requests and makes queries to local and district wrappers to retrieve data about any distributed health record of a citizen (Fig. 5). It returns the addresses of DocumentHandler agents which the client must contact to get the required documents. To accomplish this task, the gateway performs two basic activities:

- When it retrieves the distributed data required to fulfill a client request, it must integrate them into a data structure, so that the client can handle a single dataset.
- When a clinical record is produced within a district for a patient belonging to another district, the former gateway must inform the latter one to make its referring DistrictDBWrapper agent register such event in its own district database.

Init.

During the starting phase of the district platform, the Init agent registers the same platform Gateway to all the active DF_Inter-district agents of the remote district platforms in the territory.

DF_Inter-district.

As we just said, it is the Directory Facilitator in which all the remote Gateways are registered. This allows a single Gateway to communicate with any other distributed gateway in the entire infrastructure.

DF_Intra-district.

This Directory Facilitator contains all the LocalDBWrapper agents registrations of the local platforms belonging to the same district.

LoginServer.

Its task is to establish a secure connection with the client that wants to access to the infrastructure to retrieve data in a specified district.

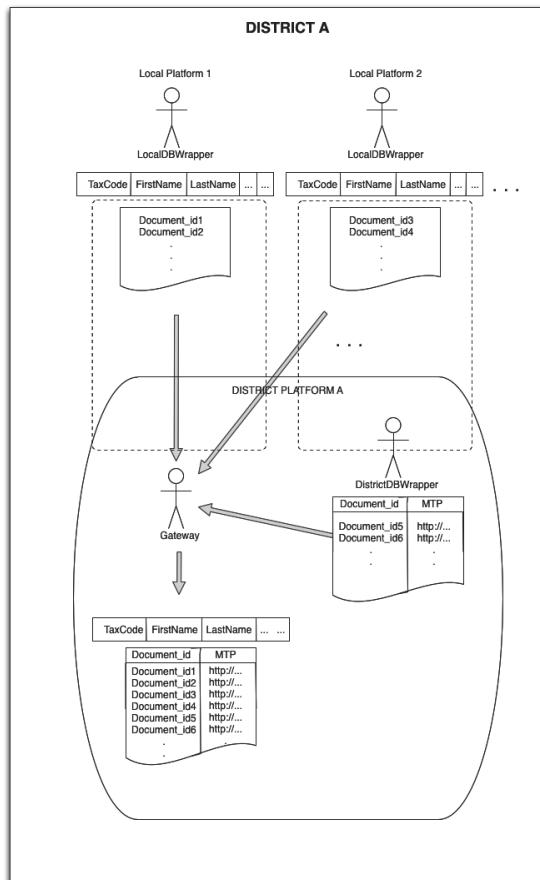


Figure 5. The Gateway agent retrieves the ubication of health records within the system.

2.3. Client Platform

This logic platform (Fig. 6) contains client applications, which may be any agent oriented software that is able, after a login phase, to access data through the connection with a district gateway agent. Examples of client applications could be: software to access Electronic Health Record (EHR), both by medical staff and citizens, mobile applications to retrieve the Patient Summary for emergency situations, software to update health records by general practitioners, etc.

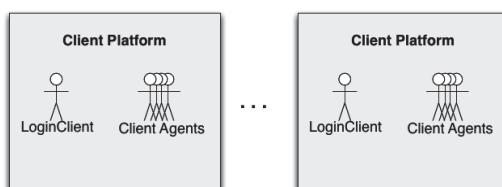


Figure 6. Client agents are any agent application that needs to access to the system.

3. Scenario

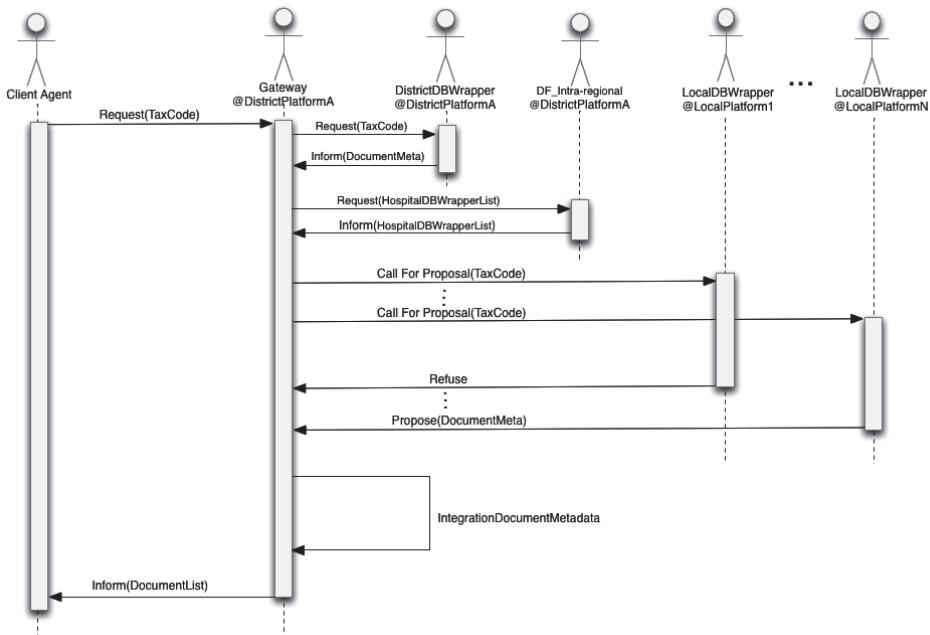


Figure 7. The client agent queries the infrastructure for a citizen's healthrecords.

To show the capabilities of this architecture we assumed a scenario where an emergency doctor urgently needs to consult a patient's health records, in particular his patient summary. According to the EU definition, a patient summary is a clinical document that is digitally stored in repositories with cumulative indexing systems and secure access by authorised people. It is an HL7 CDA compliant document, contained in the patient's EHR, whose purpose is to summarize a patient's clinical history and his current situation.

In short, the main Patient Summary's use cases can be summed up in [11]:

- Emergency situations in which the patient may not give an exhaustive description about his clinical history (problems, allergies, current medicines, etc.).
- Reliability of the information flows between family doctor and health facilities.
- Patients affected by chronic diseases managed by several specialists or elderly in home care regime.
- Diagnostic process support, telemedicine, etc.

Finally, the Patient Summary contains both mandatory and optional fields, and it is expressed through XML markup language.

To build such scenario we used:

- Jade Framework [12] to develop local and district agents in some desktop computers.

- An android smartphone application to simulate the client agent, developed with Jade Leap addon.
- Ministerial directives to compose a Patient Summary for our experiment, an XML parser and an agent ontology based on HL7 concepts.

The operating mode is very simple (Fig. 7). First of all, the mobile client application log in to the district platform entering its username and password: a secure connection is established with the platform using TSL protocol to ensure secure access to patients' personal and sensitive data. Then, the client asks for a citizen's Patient Summary and its relative health records by typing his tax code: the Gateway agent will query the different distributed entities to find the ubication of required data and inform the client where it can retrieve health records. Finally, the client application gather this data asking directly to DocumentHandler agents of the platforms which hold the patient's records.

4. Conclusion

In health information systems, the importance of addressing interoperability issues among existing systems is widely recognized. A crucial aspect is to allow health professionals to get any information they need about a patient in a pervasive and reliable way, even if these data are distributed in technically and geographically different health information systems.

To meet these requirements, in this paper we proposed an architecture based on multi-agent systems technology that takes advantage of the adoption of established standards for the management of clinical documents. Our goal was to show how MAS features can improve HIS in terms of interoperability, reliability, modularity and robustness; and how health professionals - and thus citizens - could benefit from this efficient distributed system.

References

- [1] U. Varshney, Pervasive healthcare, *Computer* **36**, 138–140, (2003).
- [2] R. Haux, Health information systems - past, present, future, *International Journal of Medical Informatics*, **75**(3-4), 268–281, (2006).
- [3] C. Doukas, T. Pliakas, and I. Maglogiannis, Mobile healthcare information management utilizing Cloud Computing and Android OS, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1037–1040, (2010).
- [4] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions, *2010 Second International Conference on eHealth, Telemedicine, and Social Medicine*, 95–99, (2010).
- [5] V. Koufi, F. Malamateniou, and G. Vassilacopoulos, Building Interoperable Health Information Systems Using Agent and Workflow Technologies, *Medical Informatics in a United and Healthy Europe*, 180–184, (2009).
- [6] D. Capozzi and G. Lanzola, An Agent-Based Architecture for Home Care Monitoring and Education of Chronic Patients, *Complexity in Engineering*, **9**, 138–140, (2010).
- [7] M. Lyell and X. Liu, Software agent application to support the patient-at-home, *International Conference on Collaboration Technologies and Systems (CTS)*, 97–103, (2012).
- [8] P. Ferronato, S. Lotti, and D. Berardi, *Strategia architettonurale per la Sanità Elettronica*, Italian Health Ministry, 2006.
- [9] L. Palazzo, A. F. Dragoni, A. Claudi, and G. Dolcini, A Multi-Agent Approach for Health Information Systems Domain, *Proc. of the 1st Workshop on Artificial Intelligence and NetMedicine*, 36–45, (2012).

- [10] M. Ciampi, G. De Pietro, C. Esposito, M. Sicuranza, and P. Donzelli, On federating Health Information Systems, *International Conference in Green and Ubiquitous Technology*, 139–143, (2012).
- [11] *Specifiche tecniche per la creazione del "Profilo Sanitario Sintetico" secondo lo standard HL7-CDA Rel. 2*, Tavolo Permanente per la Sanità Elettronica, Italian Health Ministry, 2010.
- [12] F. L. Bellifemine, G. Caire and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Wiley, 2007.

Web Service Composition with Petri Net and Ontology

Azizbek MARAKHIMOV^a, Jaegeol YIM^{b,1} and Jaehun JOO^c

^a*Cooperative Department of Techno Management, Dongguk University, Korea*

^b*Department of Computer Engineering, Dongguk University, Korea*

^c*Department of Information Management, Dongguk University, Korea*

Abstract. A web service is a function that can be invoked anytime by any program on the Internet. Making use of web services, we can very efficiently develop a new software system. However, automatic composition of web services is challenging because independently developed web services are not always compatible. Many studies suggest mediation as an effective method of composing heterogeneous web services. Among them, Petri net and ontology-based mediation is eye-catching. This paper introduces a web service composition solution that builds Petri net model of interactions between web services and ontologies representing domain knowledge, then combines the Petri net models with ontology-based mediation.

Keywords. Web Service, Composition, Petri Net, Ontology

1. Introduction

As the popularity of Service Oriented Architecture is rapidly increasing, the web service composition is becoming an emergent issue for researchers and practitioners. However, the main challenge in web service composition is the autonomy and heterogeneity of the web services. Independently developed web services are not always exactly compatible and therefore cannot be straightly composed together. Notably, in this study we do not consider the use of direct composition, as it is very primitive and ineffective method for creating real life service compositions.

Many studies suggest mediation as an effective method for composing heterogeneous web services. According to Yu et al. [1], web service mediation enables a service requester to connect to a relevant service provider regardless of the heterogeneities between them and works in a transparent way – neither of them needs to be aware of its existence. Taking into account the use of mediation in previous studies, we consider this method as a primary way for implementing web service composition. However, in order to achieve deep integration of heterogeneous web services it is important to address the semantic context of the web services, i.e. identifying semantically similar objects and solving their schematic differences. Therefore, this paper introduces ontology based web service composition solution.

¹Corresponding Author: Jaegeol Yim, Department of Computer Engineering, Dongguk University, 707 Seokjang-Dong Dongguk University, Gyeongju, Gyeongbuk, 780-714 Korea; E-mail: yim@dongguk.ac.kr

2. Related Studies

There is variety of languages for specifying web service composition such as Web Service-Business Process Execution Language for Web Services (WS-BPEL or BPEL), Web Service Choreography Interface (WSCI), and Web Service Choreography Description Language (WS-CDL). However, they are all used for direct composition of web services. We consider indirect composition, i.e. composition of heterogeneous web services with different interfaces. Tan et al. [2] refer to this as partial compatibility. Partial compatibility is the state when two or more web services provide complementary functionality and could be linked together in principle; however, their interfaces and interaction patterns do not fit each other exactly. Therefore, authors made two assumptions regarding partial compatibility:

- Incoming and outgoing messages of two given web services are different from each other;
- Two given web services do not have same message format and exchange sequence;

In order to solve the problem of partial compatibility, Tan et al. [2] provided mediation-aided composition. Mediator between two web services wraps them and solves the differences between their interfaces so that they can appear homogeneous and compatible with each other. This is an economic and labor efficient to address the challenge of partial compatibility.

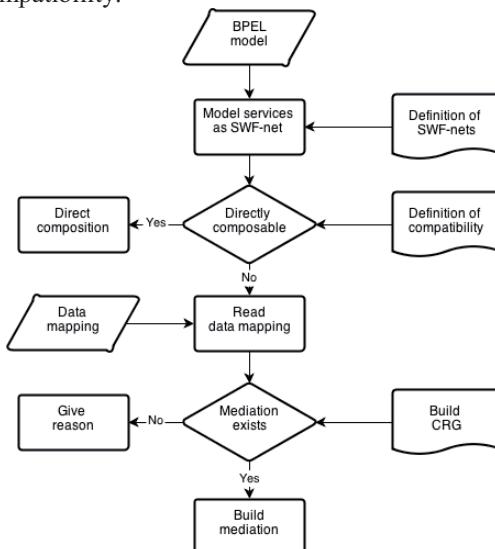


Figure 1. System architecture by Tan et al.[2]

A significant part of the paper is dedicated to transforming BPELs into colored Petri-nets. This is important phase of the solution provided by the authors. Figure 1 represents a solution provided by Tan et al. [2] to complete mediation-aided composition of two web services. As it can be seen, initial phases involve transforming BPEL models into colored Petri-nets and then checking the compatibility. The next important phase of this solution is data mapping. Data mapping is applied to define rules to relate (syntactically/semantically equivalent) elements of two messages, or two

interfaces belonging to different services, and it can be at message level, part level or element level. The data-mapping example is provided in Table 1.

Table 1. Data Mapping Table

Source	Target
eBay.Order	TPC.COREq.Order
"eBay"	TPC.COREq.PartnerID
eBay.Order.(UserID, SecretID)	eBay.(UserID, SecretID)
TPC.(OrderID, UserID)	eBay.(Token, OrderID, UserID).(OrderID, UserID)
eBay.Token	eBay.(Token, OrderID, UserID).Token
eBay.OrderData	TPC.OrderData

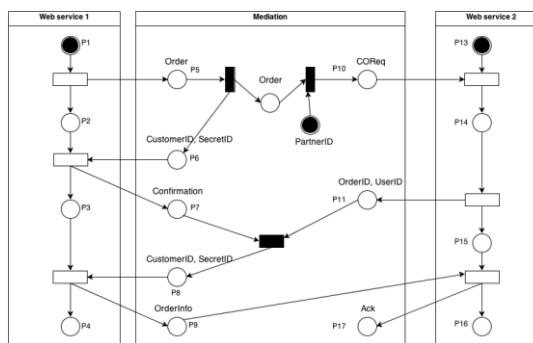


Figure 2: Mediator aided composition by Tan et al. [2]

With the given data mapping, Tan et al. [2] constructed a mediator between eBay web service and third-party checkout service. As it can be seen, two different interfaces in the data mapping eBay.Order – TPC.COREq.Order are integrated via combination of split and merge mediators. The main contribution of this paper is that it effectively addresses the problem of analyzing inter service compatibility and automatic composition of web services with the minimum engineering cost.

As the previous paper does not consider semantic context of web services and specifically the use of ontologies, the study by Cardoso and Sheth [3] was found very useful. This paper discusses semantic interoperability issues of heterogeneous web services. The authors provided a solution for ontology-based discovery and composition of web services. They used ontologies to initially describe web service interfaces. Ontology is defined as an explicit specification of knowledge conceptualization. An ontology $\Omega_i = \{c_1, \dots, c_n\}$ contains a set of classes. Each class c_j has an associated set of properties $P_k = \{p_1, \dots, p_m\}$. Each property has a range indicating a restriction on the values the property can take. In the implementation of [3], tasks and Web service interfaces are semantically described by concepts (classes) that are defined in ontologies constructed with DAML+OIL. With the ontologies, they developed a mechanism that automatically computes the similarity of two services, efficiently and without human intervention, and that suggests potential mappings.

Li et al. [4] also proposed various mediator patterns to generate executable mediators and compose partially compatible services. Their approach of mediator generation involves applying heuristic technique for identifying protocol mismatches between given web services, and then selecting appropriate mediator patterns. Li et al. [4] classified web service mediation into signature and protocol mediation. Accordingly, signature mediation mainly focuses on message types of web services. Protocol mediation aims at reconciling mismatches of message exchanging sequences.

There are several types of mismatches in web service composition such as Mismatches of extra messages, Mismatches of missing messages, Mismatches of splitting messages, Mismatches of merging messages, Mismatches of extra conditions, and Mismatches of missing conditions. For example, authors described mismatches of missing messages, as the provided interface does not have some messages that the required interface expects to send/receive.

3. Our Composition Method

Our method is basically a combination of the existing approaches, Petri net method [2] and ontology method [1, 3].

3.1. Proposed Framework and Algorithm

For our study, the system architecture shown in Figure 3 is proposed. In a similar way as previous studies, we will transform BPEL models into colored petri-nets. There are many tools to manipulate with CPNs easily. The main phase of the system is mediation generation. At this phase, similar to Li et al. [4], we create mediation workspace based on Java Eclipse environment. Mediator pattern repository will store common types of mediators. Ontology repository will store domain ontologies to specify the objects in data mapping. Consequently, mediator generation phase will consider both semantic data mapping and existing mediator patterns to analyze compatibility and select appropriate mediator types.

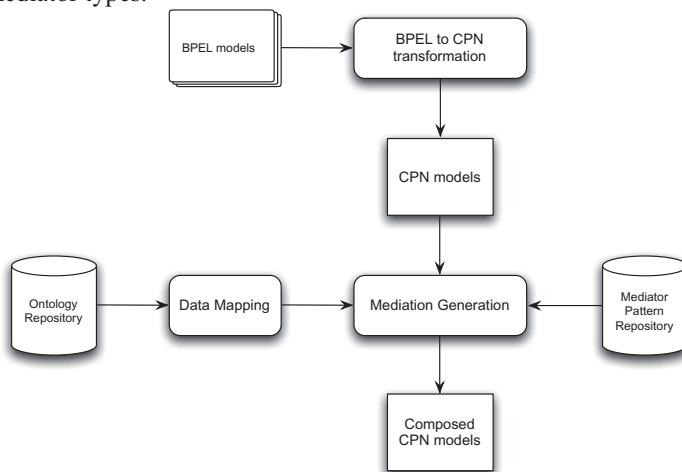


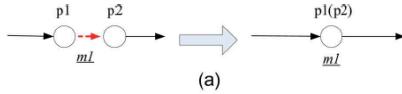
Figure 3. The architecture of our system

So far, we have developed an initial algorithm for selecting mediator type. However, this is an initial attempt to develop the algorithm for our study, and consequently it will be refined and modified further. This algorithm checks data mapping rules and selects correct mediator types. Accordingly:

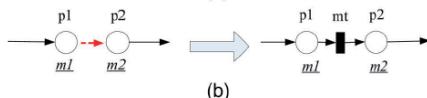
Input: Data mapping rules

Output: Mediator pattern

- If source (mr1) = target (mr1) AND target (mr1) = 1
Then use store/forward mediation;



- Else if source (mr1) != target (mr1) AND target (mr1) = 1
Then use transformation mediation;



- Else if source (mr1) = target (mr1) AND source (mr1) > 1
Then use split mediation;
- Else if source (mr1) = target (mr1) AND target (mr1) > 1
Then use merge mediation;
- Else if source (mr1) != target (mr1) AND source (mr1) > 1
Then use split mediation with transformation;
- Else if source (mr1) != target (mr1) AND target (mr1) > 1
Then use merge mediation with transformation;

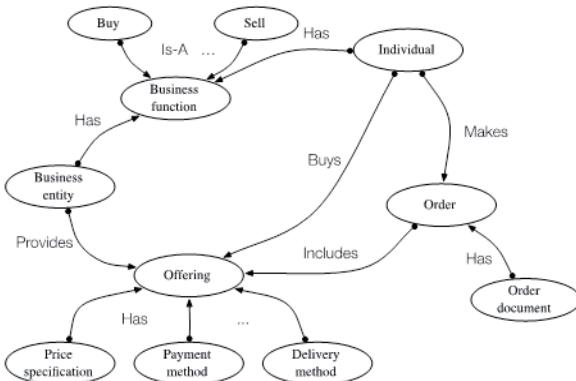


Figure 4. Abstracted domain ontology

3.2. E-commerce Ontology

The main limitation in Web service composition is that checking the compatibility of involved services requires a manual effort. To solve this problem and semantically enrich the mapping rules given above, we developed domain ontology. As eBay operates in e-commerce field and enables user to make online purchases, a widely used GoodRelations ontology was appropriate for our case. Hence, we modified it by shortening its structure and adding few new concepts. An abstract view to ontology is provided in Figure 4. We developed our ontology using TopBraid Composer software tool. We explicitly described meanings of concepts related to e-commerce activities

and properties (relationships) between those concepts by Web Ontology Language (OWL).

Specifically, *Order* concept that is used to describe online orders made by users possesses *Has* relationship with *Order* document class. *Order* document class includes subclasses as *Quote*, *Response*, *Request*, and *Receipt*. Message formats for those subclasses are also described. For example, *UserID* of the customer who made the order is one of the fields of the message. We can also notice that *UserID* can be replaced by *CustomerID* depending on the matter of taste. During annotation process all input and output messages of eBay and TPC Web services will be described using the concepts of ontology and new semantic data mapping rules will be constructed. This will enable us to automate the process of checking Web service compatibility for building mediators.

In Figure 5, we provide UML diagram of the ontology and screenshots from TopBraid Composer for better representation of classes and properties.

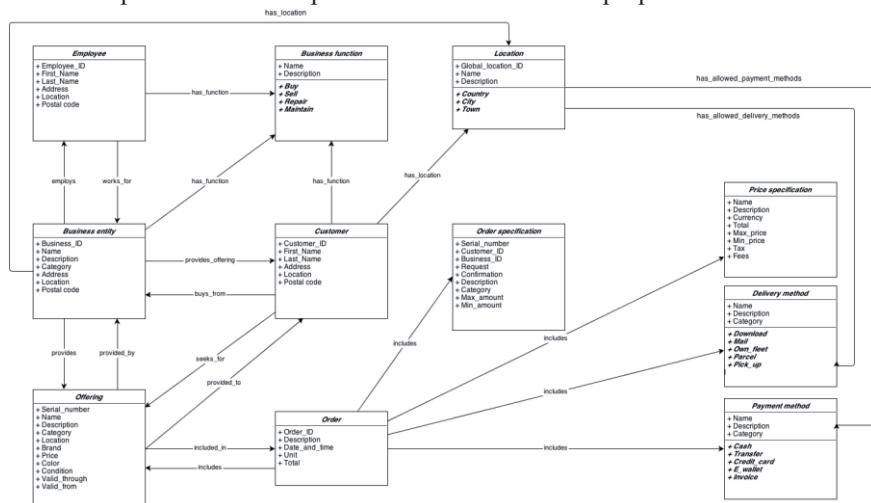


Figure 5. Ontology UML diagram

3.3. Mediation

Figure 6 shows the interaction of two Web services provided in Petri-net formalism. We adopted the motivating scenario from Tan et al [2] with the assumption that there will be several semantic heterogeneities in mapping rules. The figure shows the input and output messages of two Web services. Accordingly, Web service 1 initiates Web service 2 by sending *OrderID*, *CustomerID*, and *SecretID*. In turn, Web service 2 initiates Web service 1 by replying with *UserID* and *SecretID*. Then, Web service 1 receives *OrderID* and *UserID* and replies with *OrderInfo*. There are some cases of semantic conflict where semantically similar objects are described by different naming. Below we will identify exact cases of semantic conflicts.

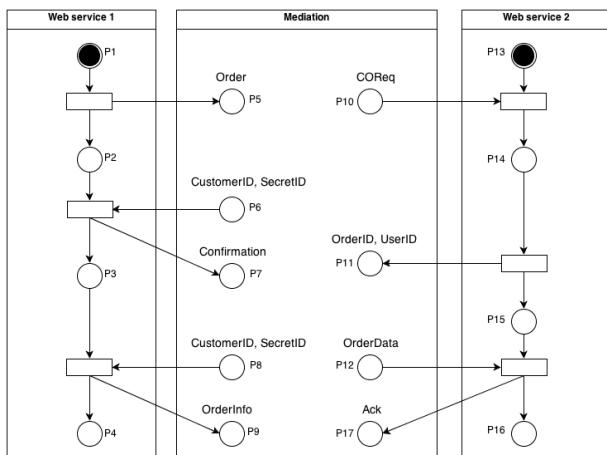


Figure 6. Annotating data messages with domain ontology

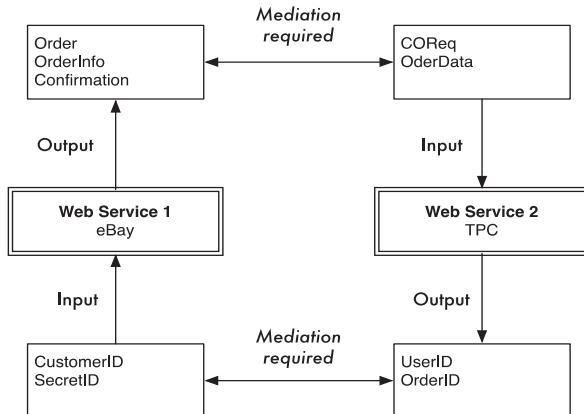


Figure 7. Need for data mediation between Web services

The motivating scenario introduced in Figure 6 represents a use of two real world Web services. We consider the process of shopping checkout of eBay site. These two Web services are from different providers that use different terminology and data formats. In many common cases, Web service B needs data from Web service A to complete the process. And the need for mediation is to convert messages from Web service A to the format required by Web service B. However, important point in our case is that both Web services need messages from each other in order to fulfill and complete the checkout process. Therefore, as we mentioned above they are partially compatible with each other. Figure 7 shows the input and output messages of both Web services. The problem is that providers used different terminology to specify semantically similar objects. This makes interfaces of two services not compatible with each other. For example, Web service 1 (eBay) sends *Order*, *OrderInfo*, and *Confirmation*, while Web service 2 (TPC) accepts *COReq*, and *OrderData*. In turn, Web service 2 (TPC) sends *UserID* and *OrderID*, but Web service 1 (eBay) accepts *CustomerID* and *SecretID*. The task that should be solved is to match outputs of *eBay*

to inputs of *TPC*, and match outputs of *TPC* to the inputs of *eBay* Web services. This situation can be solved by either conventional or semiautomatic ways. Conventional approach involves manually mapping inputs and outputs of Web services. However, this approach will limit the solution only to two selected Web services. On the other hand, we have proposed wider system framework above that includes repository of domain ontologies and mediator patterns. Our domain ontology for e-commerce is activated and performs the data mapping. Such framework will allow using other Web services or other ontologies with less manual work.

In our case, we face the problem of semantic heterogeneities where different descriptions are used for semantically similar objects. Specifically, we have semantic naming conflicts in matching inputs and outputs of two Web services. One of such conflicts is shown below:

eBay [*CustomerID*]

TPC [*UserID*]

There are few other similar conflicts that need to be resolved. As it can be seen, two different names are used to describe the identification number of customer. In Table 2, we provide full list of semantic conflicts between *eBay* and *TPC* Web services and show the way to resolve those conflicts by applying domain ontology.

Coming out from the data mapping rules, we identified several semantic conflicts. It can be seen from Petri-net model in Figure 6 that Web service 1 sends *Order*, while Web service 2 accepts *COReq* message. Also, Web service 1 sends *OrderInfo*, but Web service 2 accepts *OrderData* though they semantically mean the same thing. Therefore, we have specified them as semantic conflicts in naming similar objects.

Table 2. Semantic conflicts between Web services (a)

Mapping rules	
<i>Output of Web service 1</i>	<i>Input of Web service 2</i>
Order	COReq
OrderInfo	OrderData
Confirmation	
Identified conflicts	
<i>Order</i> vs. <i>COReq</i> – naming conflict	
<i>OrderInfo</i> vs. <i>OrderData</i> – naming conflict	

Also, we identified semantic conflicts between output of Web service 2 and input of Web service 1 in Table 3 in a similar way. As it can be seen, Web service 2 uses *UserID* and *OrderID* to describe the same objects as *CustomerID* and *SecretID* do.

Table 3. Semantic conflicts between Web services (b)

Mapping rules	
<i>Output of Web service 2</i>	<i>Input of Web service 1</i>
UserID	CustomerID
OrderID	SecretID
Identified conflicts	
<i>UserID</i> vs. <i>CustomerID</i> – naming conflict	
<i>OrderID</i> vs. <i>SecretID</i> – naming conflict	

For annotating output and input messages leading to semantic conflicts we refer to [5]. According to [5], Semantic Annotations for Web Service Description Language (SAWSDL) provides a proper mechanism to annotate the capabilities and requirements of Web services (described in WSDL) with semantic concepts defined in domain ontology. SAWSDL is based on an earlier work WSDL-S (WSDL-S), a W3C member submission for Semantic Web services. Using XML extensibility elements and attributes, semantic annotations on WSDL elements (including inputs, outputs and functional aspects like operations, their preconditions and effects) are achieved by referencing semantic concepts from one or more external domain models (ontology).

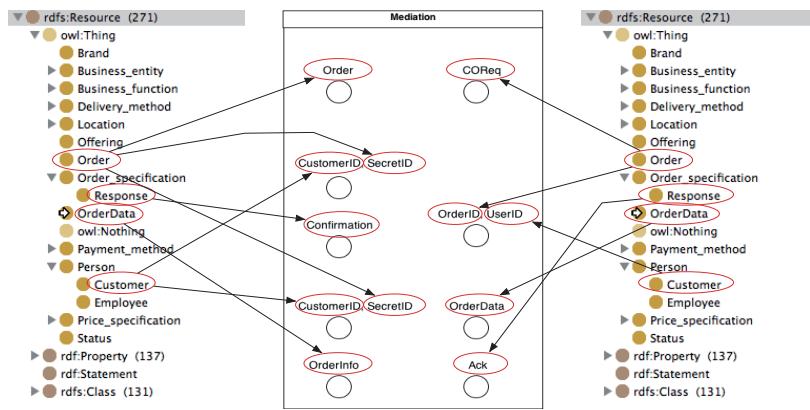


Figure 8. Annotating data messages with domain ontology

As an example of annotation for resolving semantic heterogeneities we can view the conflict between *CustomerID* and *UserID*. In Figure 8, we have annotated both of them by referring to class *Customer*. When we inquire class *Customer* from the TopBraid Composer we can get the information shown in Figure 9:

[Property]	[Range]	[Domain Class]
CustomerID	xsd:integer	Customer
Description	xsd:string	Person
First_name	xsd:string	Person
Last_name	xsd:string	Person
Seeks_for	Offering	Customer
UserID	xsd:integer	Person
has_Business_function	Business_function	Person
makes_order	Order	Customer
orders_from	Business_entity	Customer

Figure 9. Properties of class *Customer*

The message lists all object and data properties that class *Customer* holds. There is *CustomerID* among other properties. Also, there is *UserID* that belongs to class *Person*, which is a parent class of *Customer*. In addition, the use of *sameAs* OWL function, as shown in Figure 10, supports interchangeable use of *CustomerID* and *UserID*, and they will be processed as semantically similar objects. In this way, we will resolve all semantic conflicts that were identified above.

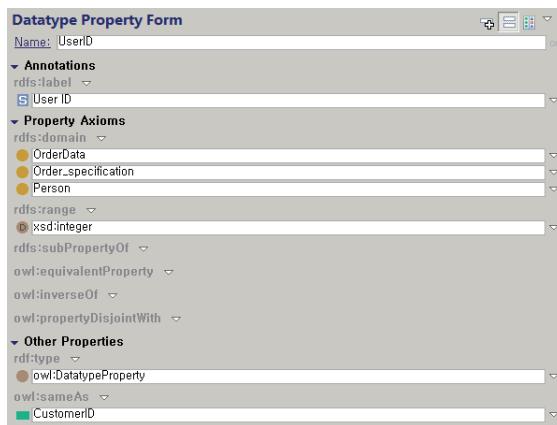


Figure 10. Assigning *sameAs* function to *UserID*

4. Conclusions

This paper proposed a new method of composing web services. It builds up Petri net models of interactions between web services and represents domain knowledge with ontology. Then, it combines web services referring to the ontology. For the future work, we are automating the proposed process of combining web services.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0002611). This work (Grants No. C0033172) was also supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

References

- [1] Q. Yu, X. Liu, A. Bouguettaya, B. Medjahed, Deploying and managing web services: issues, solutions, and directions, *The International Journal on Very Large Data Bases (VLDB Journal)*, 17 (2008), 537–572.
- [2] W. Tan, Y. Fan, M. Zhou, A petri net-based method for compatibility analysis and composition of web services in business process execution language. *IEEE Transaction of Automation Science and Engineering*, 6 (2009), 94-106.
- [3] J. Cardoso, A. Sheth, *Semantic e-workflow composition*. LSDIS Lab, University of Georgia, 2002
- [4] X. Li, Y. Fan, S. Madnick, Z. Q. Sheng, A pattern-based approach to protocol mediation for web services composition. *Information and Software Technology*, 52 (2010), 304-323.
- [5] Y. Shinjo, T. Kubo, C. Pu, Efficient mediators with closures for handling dynamic interfaces in an imperative language, *Information and Software Technology*, 46 (2004), 351–357.

A New BDI Architecture To Formalize Cognitive Agent Behaviors Into Simulations

Van-Minh LE^{a1}, Benoit GAUDOU^b, Patrick TAILLANDIER^c and Duc-An VO^a

^a*IFI, Equipe MSI; IRD, UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique, Vietnam National University, Hanoi, Vietnam*

^b*UMR 5505 IRIT, CNRS, University of Toulouse, France*

^c*UMR IDEES, CNRS, University of Rouen, France*

Abstract. Nowadays, agent-based modelling is more and more used to study complex socio-ecological systems. These last years have also seen the development of several agent-based simulation platforms. These platforms allow modelers to easily and quickly develop models with simple agents. However, socio-ecological systems need agents able to make decisions in order to represent human beings and the design of such complex agents is still an open issue: even with these platforms, designing agents able to make complex reasoning is a difficult task, in particular for modelers that have no programming skill. In order to answer the modeler needs concerning complex agent design, we propose a new agent architecture based on the BDI paradigm and integrated into a simulation platform (GAMA). This paradigm allows designing expressive and realistic agents, yet, it is rarely used in simulation context. A reason is that most agent architectures based on the BDI paradigm are complex to understand and to use by non-computer-scientists. Our agent architecture answers this problem by allowing modelers to define complex cognitive agents in a simple way. An application of our architecture on a model concerning forest fire and fire-fighter helicopters is presented.

Keywords. Cognitive agent design, BDI architecture, simulation platform

1. Introduction

The use of agent-based simulation to study socio-ecological systems is booming since twenty years: for example DeAngelis' work in ecology [1], Tesfatsion's in economy [2] or Gilbert's in social sciences [3]. In fact, agent-based simulation is a powerful tool to study complex systems, in particular when these ones deal with human beings. However, in most of actual models, the modeling of the human behavior is still rather simple and this has a negative impact on the model realism. A reason is the difficulty to design cognitive agents.

¹ Corresponding Author: Van-Minh LE, IFI, Equipe MSI; IRD, UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique, Vietnam National University, Hanoi, Vietnam; E-mail: vanminh.le246@gmail.com

These last years have seen the emergence of several platforms easing the development of agent-based models (NetLogo [4], GAMA [5]...). However, even with these platforms, the problem of the cognitive agent design is still an open issue. In fact, designing agents able to make complex reasoning is a difficult task, in particular for researchers that have no programming skill.

In this paper, we propose new cognitive agent architecture based on the BDI (Belief, Desire, Intention) [6] paradigm and directly integrated into the GAMA simulation platform. Our architecture allows modelers to define complex agents and at the same time to understand and to use them easily.

The paper is organized as follows. In Section 2, we present a brief state of the art of agent architectures. Section 3 is dedicated to the presentation of our BDI agent architecture. In Section 4, we present an application of the architecture in a context of a model concerning helicopter firefighters. At last, Section 5 concludes.

2. State of the Art

The problem of the agent design is a classic problem in agent-based simulations and numerous formalisms and architectures were proposed. Among all these formalisms or architectures, the finite state machines or the motivational architecture [7] can be very useful when designing simple agents, but are not adapted to complex cognitive agents as their representation capability is fairly limited.

A classic paradigm to formalize the behavior of cognitive agents is the BDI (Belief, Desire, and Intention) paradigm [6]. This paradigm allows designing expressive and realistic agents. It has been implemented in a huge number of architectures. Most of existing architectures are based on Bratman's resource-bounded reasoning principle [8,9] and on the PRS (Procedural Reasoning System) framework [10].

The PRS framework [10] makes the assumption that an intelligent agent should have a thinking process before reacting. This framework includes three main processes: the perception (in which agent acquires information from the environment), the central interpreter (which helps the agent to deliberate its goals and then to select the available actions) and the execution of intention (which represents agent's reactions).

The resource-bounded reasoning concept proposed in [8] makes the assumption that the thinking process costs resources (at least computation-time). In order to cope with this issue, Bratman proposes to use plans. In fact, an agent does not have to take into account all the perceptions and actions, but it only has to consider a set of pre-defined plans that depend on the current situation.

The idea of using plans was implemented in the AgentSpeak architecture (JASON) proposed by Bordini [11]. In this architecture, an agent begins its planning phase only when a trigger comes from a change in the agent's mental states. Besides, no desire is considered in the planning phase.

Another classic "plan-based" BDI architecture is JAM (proposed in [12]). In contrary to the AgentSpeak architecture, plans in JAM contain the desires as goals to pursuit. In this architecture, the agent's behaviours are implemented through the plans: agents are more reactive than cognitive.

Some studies have shown the interest of using the BDI paradigm in simulation context (e.g. [13] or [14]), yet, it is still rarely used. A reason is that most agent architectures based on the BDI paradigm (e.g. JAM and JASON) are complex to understand and to use by non-computer-scientists.

3. Architecture

3.1. Architecture Overview

A global overview of our architecture is presented in Figure 1. Each part is described in details in the sequel. Our architecture is based on both the PRS framework [10] and the resource-bounded reasoning principle. We propose to use the same type of event-based approach as JASON one [11].

3.2. Description of the BDI Components

Beliefs represent the informational states of the agent. In BDI logic, $Bel_i(\varphi)$ expresses the fact that the agent i believes φ . In our proposition, the belief is directly associated with the agent, so the agent i is omitted from the description of a belief. The content of the belief expresses a state or an activity concerning the agent or its world. We propose to use a triplet (*subject*, *predicate*, *object*) to describe this content. For example: $Bel(helicopter1, takeCare, tree12)$ expresses that the agent has a belief that agent helicopter1 will take care of the agent tree12. While the formula $Bel(X, burning, null)$ means that the agent believes that agent (tree) X is burning (i.e. in the state burning), the $Bel(X, Bel(tree12, burning, null), null)$ declares that the agent believes that the agent X believes that the agent tree12 is burning.

Desires express the motivational states of the agent. As desires are also mental states like beliefs, a desire shares with a belief the same content format. From Bratman's idea that an agent cannot simultaneously pursue competing desires, we propose two additional attributes to describe a desire:

Competing category: each desire belongs to a competing category. Two desires of the same category cannot be considered at the same time (except for the desires belonging to no category).

Priority: the priority is the degree of importance of a desire. The higher the priority, the more important the desire is. Among the desires of a competing category, the agent chooses the desire with the highest priority.

Events trigger the reactive activity of the agent. In the AgentSpeak architecture, an agent begins reacting when its mental states change. In this architecture, events are described as a creation or a deletion of a mental state. It means that when an agent acquires a new belief or a new desire, this agent creates an event. For example, when an agent perceives a new burning tree (tree12), it first gets the belief and then an event corresponding to this new belief creation is triggered. In our architecture, we reuse the format of event proposed by Bordini: a belief creation event is for example represented by $Bel(+, tree12, burning, null)$.

Rules are what an agent uses to make the logic deduction creating new beliefs and new desires from current beliefs and desires. The modifications of mental states come not only from what the agent perceives but also from its internal reasoning process. That means that an intelligent agent is also capable to reason in order to update its beliefs and desires according to its current mental states. As the first order logic is used to describe the beliefs and desires, we propose to use the implicative normal form to describe the rules. As example, the following expression means that an agent would create the desire to go toward another agent (tree agent) if it perceives that this agent is burning and if it is carrying water:

$$Bel(_self, carrying, water) \wedge Bel(X, burning, _null) \Rightarrow Des(_self, goto, X)$$

A **Plan** is a sequence of declared actions that the agent has to apply to reach one (or many) goals. This means that a plan describes the fact that the agent has to execute some particular actions once it gets a specific condition on mental states (beliefs, desires). A plan is composed of goals (desires of the agent), context (conditions on mental states), trigger (events that trigger the plan) and actions (actions to execute).

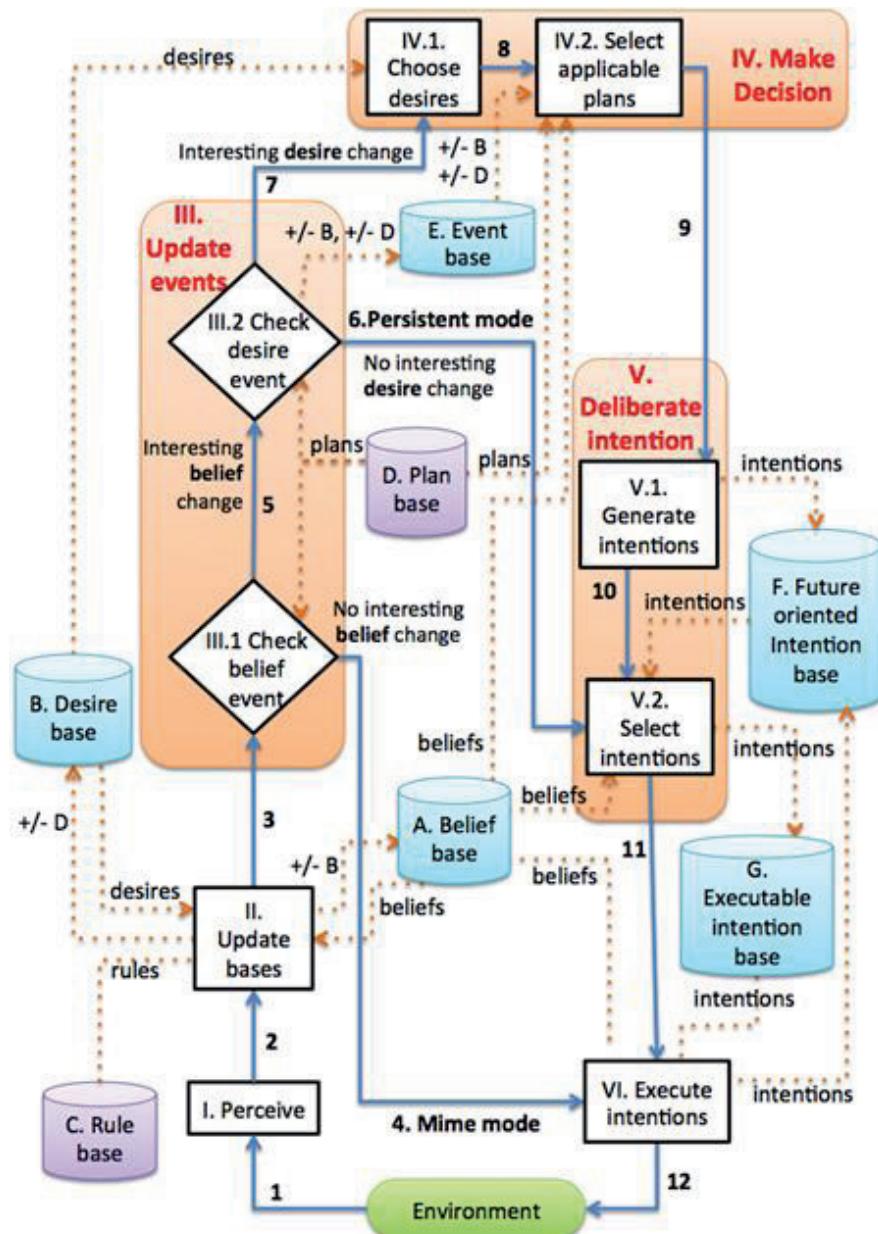


Figure 1 – BDI architecture

The following plan declares that the agent will execute the action `.informFire(X)` (to tell other agents that the agent X is on fire) and then the action `.goto(X)` (to move to the agent X) to satisfy the desire to extinguish the fire on the agent X when it is carrying water and it gets the information that the agent X is burning. The agent begins to reason on this plan only when it gets the new belief that an agent X is burning.

Plan:

```
goal (Des_ self, extinguish, X));
context (Bel(X, burning, null), Bel(_self, carryingWater, null));
trigger (Bel(+, X, burning, null));
action (.informFire(X), .goto(X));
```

When all the conditions of a plan (the goal, the context and the trigger) are satisfied – which means that the agent has the beliefs, the desires and the events matching with the all the variables of the plan conditions – the plan is considered as applicable.

The **Intention** represents the deliberative state of the agent, i.e. what the agent has chosen to do. According to Bratman, “intending to do something (or having an intention) and doing something intentionally are not the same phenomenon” [15]. Thus, intentions are classified into two types:

A Future-oriented intention is a specific instance of an applicable plan.

A Present-oriented intention is the future-oriented intention that the agent has chosen to pursue.

Actions are one of the components of an intention. An intention may contain several actions that will sequentially be performed. Each action describes the agent’s behaviour and the action conditions. An action is composed of three components corresponding to three situations:

Normal situation is a situation where the mental condition of the agent meets the intention condition that is inherited from the plan context. At this moment, the action is normally performed.

Success situation is a situation where agent’s mental condition allows the agent to decide that certain goals are achieved (for example, the agent acquires new desires that allow the agent to satisfy the desires of its intention goals).

Failure situation is a situation where the agent considers that the action failed and where it decides to stop following the action goal. In this case, the agent removes the corresponding desires or creates new events that trigger the backup plans.

3.3. Practical Reasoning Processes

Perception: The agent either perceives a change in its environment or receives a message from other agents (processing flow 1). The agent then subjectively considers this new information to select what it finds useful by using a user-programmed filter. The filtered information is transferred into its beliefs. For example, in the context of a forest fire simulation, a helicopter agent may simultaneously perceives burning and intact trees; in the this simulation context, intact trees are not important, so they are ignored; in contrary, the burning trees are important, so the beliefs on these trees (for example: `tree12, tree13`) are created as follows: `Bel(tree12, burning, _null)`; `Bel(tree13, burning, _null)`.

Update beliefs and desires: In this phase, the agent executes the logic deduction on the current beliefs and desires (including the new incoming beliefs) by applying the rules to create new beliefs and desires (processing flow 2). Once new beliefs or desires are created, agent adds the corresponding events into the event base. At the end of this phase, the agent obtains new beliefs (added to the belief base), desires (added to the desire base) and events that will be used in next phase.

Update events: In this phase, the agent analyses the events created in the previous phase (processing flow 3) in order to choose a suitable behaviour according to the current context. The idea here is that the agent only needs to elaborate a new plan if the changes from its world impacts what is already planned. So instead of losing time to choose again the same plan, the agent continues what it is doing. The agent begins planning only when there are new interesting events. An event is considered interesting when it matches one of the abstract events described in the trigger part of the plan. The events can be of two types: belief events and desire events. The filtering of events is thus separated into two steps.

Step 1: Verify interesting events on beliefs. The agent checks whether it perceives any interesting events of belief. If there is no belief event considered as interesting, the agent considers that there is no change from its environment or that the changes are not important enough to force it to reconsider its situation. In this context, the agent executes the current executable intentions (processing flow 4). The fact that the agent ignores the thinking process to continue pursuing its intention is called mime mode, which means that the agent repeats what it was already doing. In contrary, if the agent perceives some interesting belief events, it moves to the next step to verify the desire events (processing flow 5).

Step 2: Verify interesting events on desires. In this step, the agent checks whether there are any interesting events of desire or not. If all the desire events are considered as not interesting, the agent keeps following the same goals and continues to select the future-oriented intentions, which are deliberated according to the goals that were previously chosen (processing flow 6). We call the way the agent continues to deliberate intentions the persistent mode, because the agent persistently pursues its pre-decided goals.

In case the agent finds that some of the desire events are interesting, it moves to next phase to decide what to do (processing flow 7).

Decision-making: In this phase, the agent makes its decisions so that its next behaviors are compatible with the current conditions of its mental state (which are corresponding to its environment modification). Concerning the decision-making process, previous works have proposed plenty of approaches on how an agent decides what to do. For example the approach proposed by Bordini (in AgentSpeak) [11] states that agent's planning phase is based on the intention.

In our architecture, we propose another approach: the agent establishes its plan according to its desires. The planning phase only consists in creating the future-oriented intentions. The actions are only created when the context condition of the intention is met. Thus, the decision in our model is separated into two steps.

Step 1: Selection of desires. From its desire base, the agent selects a sub-set of desires according to their competing category and priority. First, the agent takes all non-competing desires. Then, among the desires of a category, it selects only the ones that have the highest priority.

Step 2: Select applicable plans. The agent selects the plan of which the conditions are met according to the agent's mental states (processing flow 8). The plans are

selected in the following way: The agent only selects the plans corresponding to the desires chosen from the previous step; The agent uses the objects and the subjects from all the beliefs and desires in its bases to apply to the variable of the abstract conditions of each plan; For each instance of the plan, the agent verifies all the conditions (including the context condition and the triggering condition); An instance of the plan of which all the conditions are satisfied is considered as applicable. The agent then uses it for the next phase (processing flow 9).

Deliberate intentions: In this phase, the agent organizes its intentions to select the current executable intention to perform. First, the agent creates the future-oriented intentions from the applicable instances of plans. The intentions, inheriting the goals, the context and the actions from the plan, are put into a structured base. In this base, these intentions are grouped according to their desire competing categories. Then, the agent chooses the intentions which conditions are currently satisfied according to the belief base (processing flow 10). The satisfied intentions are also called present-oriented intentions and are put into the executable intention base for the next phase (processing flow 11).

Execute intentions: In this phase, the agent performs one by one the actions of the selected intentions in order to react to its environment, and then it verifies the predefined situation condition of each action. If the success condition is satisfied, the agent performs the behaviors for success situation. If the failure condition is met, the agent executes the behaviors for failure situation. In both cases, the intention is removed from the base. In the normal situation, the agent keeps the intention into its base for the next step of the simulation.

4. Application and Discussion

4.1. Implementation and Evaluation

In this section, we present an application of our architecture on a simulation of forest fire. In the simulation, we model the spreading of fire in a forest of 250x150 m² size in which there are 1000 agents representing the tree (which have the 3 states of the real tree in case of fire: intact, caught fire, dead), 2 agents representing the firefighter helicopters (which have the 3 simple actions: patrolling, extinguish fire, recharge water) and 1 agent representing the base where the helicopters recharge the water.

In order to evaluate our architecture, we built two strategies: one based on our new BDI architecture and another one based on simple reactive behaviors. For the BDI-using strategy, we built the plan base and the rule base so that firefighter helicopter could pass the maximum trees instead of moving to the nearest burning tree as described in the simple strategy.

The chart in Figure 2 shows that the BDI-using strategy is more effective than the reactive one to extinguish fires, which shows the usefulness of our proposed BDI architecture to make better strategies. In fact, this result chart comes from the average result of 100 testing times of simulations for both strategies. Most of the simulations return a stable result which shows that the performance of BDI-based strategy is approximately two times better than the reactive strategy.

4.2. Discussion

In addition to allow modelers to build rich and effective behaviors, our architecture also allows them to easily modify the agent behavior:

The structure of the plan permits to easily build plenty of backup plans. Modelers only need to define various plans with the same goal and each plan can have distinct context conditions. The reasoning phase would help the agent to select suitable plans according to its specific situation.

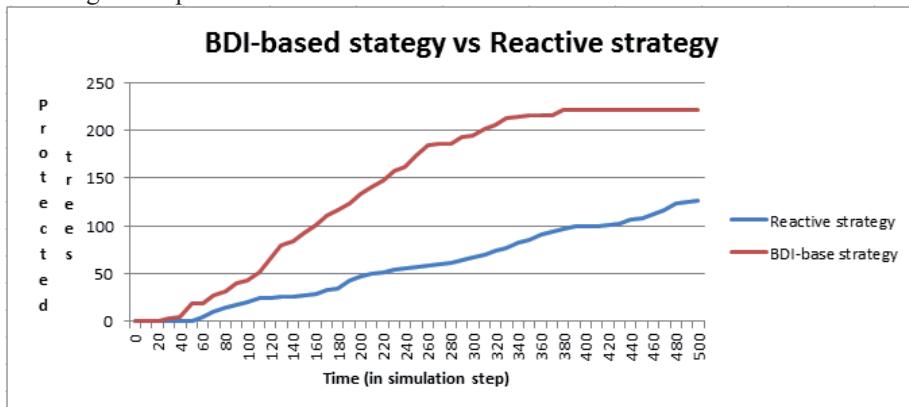


Figure 2 – Result of testing BDI-based strategy and Reactive strategy

The agent behavior strategy is based on a pre-defined plan base and on a rule base. Thus, modeler can easily fix agents actions, and then change these two bases to test different strategies.

5. Conclusion

In this paper, we propose a new cognitive agent architecture based on the BDI paradigm. Our architecture allows defining easily cognitive agents able to make complex reasoning. We illustrated the use of our architecture through a simple model dealing with forest fire and firefighter helicopters.

Although our framework have been tested on a simple simulation, the way we represent the knowledge and the decision-making process proposes promising extensions so that our architecture would adapt to other problems. An example of theoretical extension is that the decision phase will be made by considering not only the importance of the desire but also the urgency level, which will make the architecture adapt to the other simulations of urgent situation.

References

- [1] D. DeAngelis, L. Gross, *Individual-Based Models and Approaches in Ecology*, Chapman and Hall, New York (1992)
- [2] L. Tesfatsion, Agent-based computational economics: growing economies from the bottom up, *Artificial Life*, vol. 8 (2002), 55-82
- [3] N. Gilbert, K.G. Troitzsch, *Simulation For The Social Scientist*, Open University Press (2005)

- [4] U. Wilensky, Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, *Northwestern University, Evanston, IL* (1999)
- [5] P. Taillandier, A. Drogoul, D.A. Vo, GAMA: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *The 13th International Conference on Principles and Practices in Multi-Agent Systems* (2012)
- [6] A.S. Rao, M.P. Georgeff, Modeling Rational Agents within a BDI-Architecture, (1991) 473-484
- [7] G. Robert, A. Guillot, A motivational architecture of action selection for non-player characters in dynamic environments, In *International Journal of Intelligent Games & Simulation*, vol. 4 (2005), 1-12
- [8] M.E. Bratman, D.J. Israel, M.E. Pollack, Plans and resource-bounded practical reasoning, *Computational Intelligence*, 4(4) (1988), 349-355
- [9] M.E. Bratman, Two Faces of Intention, *The Philosophical Review*, Vol. 93, No. 3 (1984), 375-705
- [10] K.L. Myers, User Guide for the Procedural Reasoning System Technical Report, *Artificial Intelligence Center, Technical Report, SRI International, Menlo Park, CA* (1997)
- [11] R.H. Bordini, J.F. Hübner, M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason*, Wiley Series in Agent Technology, John Wiley & Sons (2007).
- [12] M.J. Huber, JAM: A BDI-Theoretic Mobile Agent Architecture. *Agents* (1999), 236-243
- [13] C. Adam, B. Gaudou, S. Hickmott, D. Scerri, Agents BDI et simulations sociales, Unis pour le meilleur et pour le pire. *Revue d'Intelligence Artificielle, Special issue: Simulation sociale orientée agent, Hermès Science Publications*, Vol. 25 (2011), 11-42
- [14] A. Shendarkar, K. Vasudevan, S. Lee, Y.J. Son, Crowd simulation for emergency response using BDI agents based on immersive virtual reality, *Simulation Modelling Practice and Theory* 16 (2008), 1415-1429
- [15] P.R. Cohen, H.J. Levesque, Intention is choice with commitment, In *Journal Artificial Intelligence archive*, Volume 42 Issue 2-3 (1990)

Agent-based Military Engagement Simulation System Architecture with Implementation

Won K. HWAM^a, Yongho CHUNG^a, Junghoon KIM^b, and Sang C. PARK^{a,1}

^a*I. E. Dept. of Ajou University, Rep. of Korea*

^b*Agency for Defense Development*

Abstract. Presented in this paper is architecture of a simulation system for the military engagement. The architecture proposed in this paper is for the construction of a simulation system based on agent simulation design methodology. Every entity of the proposed architecture is defined of assemblage of modularized agent components, and it enhances the reusability and composable of the entity. Consequently, time, costs, and efforts that are required to develop a new simulation system are able to be reduced by the improved entity modeling methodology of the architecture. In the case of the military engagement simulation, reflection of environmental effects to simulation entities is very important, because real battle entities are operating in the battlefield of which cannot be controlled by operators. The proposed architecture contains the synthetic battlefield to simulate interaction between the environment and simulation entities. Thus, each behavior of simulation entities is changeable by the effects, and it can derive more precise results that well-represent real system operation. In conclusion, the proposed architecture supports swift construction of various engagement simulation systems that are based on the synthetic battlefield. The architecture of this paper is implemented to construct an example system for verification of the advantages.

Keywords. Agent, Architecture, Engagement, Military, Simulation, Synthetic Battlefield, System Design

1. Introduction

Future battlefield concept and combat paradigm have been changed along the remarkable progression of weapon system development, and it caused military forces face unprecedented and various tactical requirements[1]. However, available military resources are very limited comparing to the requirements. The phenomena encourage adopting Defense Modeling and Simulation (DM&S) in the military forces that intend to develop new weapon systems. DM&S brings the maximum effects with the minimum costs by verification of a designed weapon system meets the requirements or not, before resources were committed[2]. Thus, many countries endeavor to apply DM&S to the acquisition cycle of weapon systems that comprises seven steps, namely, development concept, design verification, prototyping, evaluation and testing, production and deployment, operations, and subsequent logistics[3].

¹ Corresponding Author: Sang C. Park, Department of Industrial Engineering, Ajou University, Suwon, Rep. of Korea; E-mail: scpark@ajou.ac.kr

DM&S is classified as campaign, mission/battle, engagement, and engineering simulation model by the detail level of the representation. The engagement model that is the objective of this paper describes minutely behaviors and functions of entities for short duration from minutes to hours, but it does not concern with tactical command relatively. Outcomes obtained from the engagement model are such as survivability, vulnerability, and detectability[4, 5]. By the characteristics of the engagement model, it describes design properties and behaviors of each entity involved in the combat and interaction among entities as well as between the entities and synthetic battlefield[6]. In the engagement, environmental effects can be a decisive factor to defeat the opposite force, and it has been common knowledge of tactics that is proved by war history. Therefore, the results of the engagement simulation must perform various scenarios of engagements of simulation entities including environmental effects, and it is the reason that the model requires to contain the synthetic battlefield[7].

Simulation systems for the engagement need to meet next three requirements that are 1) the modeling methodology that constructs every single complex weapon system as combination of unit systems, 2) composability and reusability to create and modify an entity, swiftly and conveniently, 3) effect and impact simulations between simulation entities and the synthetic battlefield. In order to meet the demands, the system should have modularized components that perform each system of a weapon system such as movement and fire control, and every weapon entity of the simulation system is generated by combination of the components[8]. The entity that made of the combination is able to attach, detach, and replace its own components, so the entity is composable and reusable[9]. The synthetic battlefield is constructed based on environmental data such as height-map for the terrain, concentration of oxygen and humidity for the atmosphere, and salinity and temperature for the underwater. Simulation entities obtain the data from synthetic battlefield and process it to reflect own behaviors.

Common existing simulation tools are developed for general purposes or combat simulations. The tools for general purposes are such as ARENA, AnyLogic, DEVS(Discrete Event System Specification), EMPlant, Timed-FSA(Finite State Automata), and the tools for combat simulations are such as VBS2(Virtual BattleSpace2), Half-Life. Although the general purpose tools can construct not only combat simulation models but also various simulation models, it consumes much efforts and time to construct. In contrary, combat simulation tools are able to construct combat simulation models without much efforts and time consumption[10]. However, the combat simulation tools allow users very limited access for the development of simulation models, so the users are unable to build exact models that they intend to. Moreover, both types of the tools are confined reflecting environmental effects[11]. Therefore, existing simulation tools cannot meet the requirements of the military engagement simulation such as various scenarios based simulations, simplified modeling of a simulation entity, reusability of created entities, and reflection of environmental effects.

The main objective of this paper is to propose architecture for the engagement simulation system, which bridges to solve the gap between the existing tools and the requirements of the tools. The proposed architecture has synthetic battlefield providing environmental data to the simulation system and simulation entities designed by agent-based simulation methodology. The agent-based simulation entity of the system is composed of various functional components, and it controls itself. Existing case studies of the agent-based simulation system construction are mostly related to the autonomy

of CGF(Computer generated force) in cases such as OneSAF(USA), WISDOM-II(Australia), and MANA(New Zealand)[12-14]. However, the cases do not cover agent-based simulation entity construction enough. Those are rather the conceptual view than the concrete engineering methodology. Moreover, there is not enough explanation of the synthetic battlefield construction. The proposed architecture of this paper focuses on the composable design methodology of the agent-based simulation entity and also on the system design for the interaction between the simulation entities and the synthetic environment. The simulation system that is constructed based on the proposed architecture can have simulation entities that are reusable, expandable, changeable, and interacting with the synthetic environment and it meets the requirements of the engagement simulation system. In this paper, there is a description of an example system as the implementation of the proposed architecture to demonstrate its utility.

The overall structure of the paper is as follows. Section 2 presents the overall approach to the architecture development, and Section 3 gives the explanation of the proposed architecture design. The implementation results of the simulation system based on the proposed architecture are explained in Section 4. Finally, concluding remarks are addressed in Section 5.

2. Technical Approach

2.1. Agent-based Simulation Entity Design

Agent-based simulation entity design constructs a simulation entity as assemblage of various modularized functional agent components which are selected from the agent component library. Each agent component is linked with an entity as plug and play. In order to design an entity that is composed of link-able agent components, this paper adopts the core/shell design methodology[15] and modifies the methodology for the engagement simulation system. The modified methodology is described in Figure 1. The core/shell based entity contains properties of the entity in the core part, such as entity ID, entity type, force side, CAD model data, weight, size, position, damage, noise, and so on. The shell part is to store agent components and to interface between agent components and the entity, so it allows agent components to access the information of the core part. Agent components which are stored in the shell part refer and modify the core part data as results of behaviors. An agent component is the representation of a system that can be movement, fire control, detection, and protection, and it is constructed with a unit behavior and an aggregation of state variables. The unit behavior is sequential operations of the agent component, and the state variables are referred by each operation. Thus, an operation is able to represent various activities by states of the variables. For example of an agent component, a movement system has maximum speed, acceleration, and gradability as the state variables and the behavior of the component performs different moving speed of an entity by the variables.

2.2. Synthetic Battlefield Construction

In order to apply environmental effects into the engagement simulation, the simulation system is required to construct the synthetic battlefield that contains numerical environmental data. The synthetic battlefield can shape several structures by

environment types. As shown in Figure 2, the representations of oceanic/atmospheric environment and terrain properties need a time-dependent spatial dimensional grid-based structure and the representation of terrain needs a height-map[16]. Numerical environmental data must be formed in the appropriate structure, and the simulation system can own the synthetic battlefield by obtaining the structured environmental data. The synthetic battlefield provides requested environmental data from the simulation system, and the system uses the provided environmental data to derive environmental effects that are applied to results of behaviors of simulation entities.

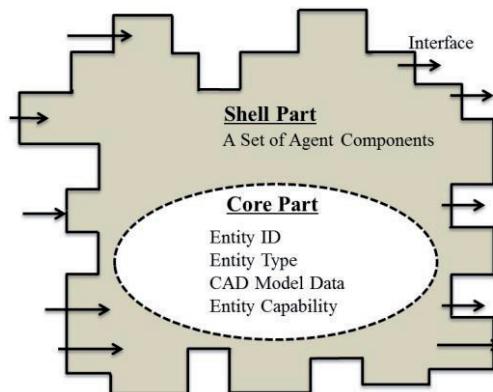


Figure 1. Core/Shell based battle entity design

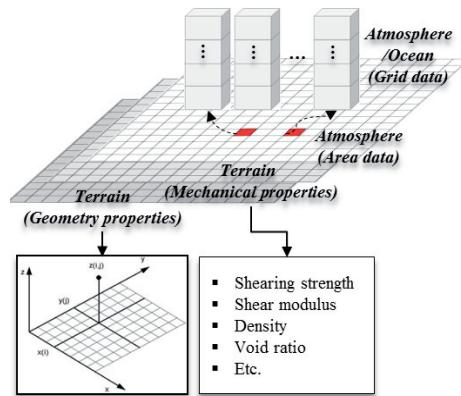


Figure 2. Structures of the synthetic battlefield

3. Architecture Design

In order to explain the system architecture that has simulation entities based on the core/shell methodology on the synthetic battlefield, this paper is along with a simple example scenario. The example scenario is for an engagement between two ground combat entities. One entity observes a specific area, and another entity goes through the area. When the observing entity found an enemy entity, it launches its weapon to attack

the enemy. By the attack of the observing entity, the moving entity can be damaged and destroyed or succeed passing the area.

According to the scenario, the simulation system needs two ground combat entities and synthetic battlefield for the representation of the terrain and the atmosphere. Each ground combat entity is designed based on the core/shell methodology, so a core part of the entity has variables for basic identity information; entity ID, type, force side, and CAD model; and variables for entity state data; weight, size, durability, current position, current speed, orientation, target position, target entity, detection distance, detected enemy entities, noise emission level, damaged level, and impact. A shell part of the entity has ports to connect agent components, such as movement, detection, fire control, and protection component. A shell part also provides an interface between a core part and agent components to allow to access and modify the variables of a core part by agent components.

The agent component that is connected to a shell part of an entity has characteristics as follows. Decision making for the agent component behavior working is dependent on the variables states of an entity core part. Agent components refer the variables of a core part, when an event is occurred for checking entity state. By the state of variables, an agent component decides working its own behavior or not. If an agent component decided working, it sends an event, which is appropriate for current entity state, to a behavior model. Activities of a behavior model refer component state variables and apply the variables to the result of activities.

Both observing entity and moving entity of the example scenario are able to be modeled as a ground vehicle entity, though they have different missions. We designed the entities with four agent components that are movement, detection, fire control, and protection, and each component design is explained in the next. First, the movement agent component compares current position with destination, and both state variables are declared in the of the entity core part. If two variables are not equal, the component starts a moving operation. During the moving operation as a behavior of the movement component, the component makes the parent entity move the specific distance that is reflected the property of the component, such as maximum velocity, acceleration, maximum rotating arc degree, gradability, and crossing depth. The state variables, which are current position, current speed, orientation, and noise emission level, of the parent entity are changed as a result of the behavior. Following pseudo code is to represent that the component operates movement, when the current position and destination are different. The moving operation of the component obtains the orientation and the moving speed, which are reflected terrain effect.

```
IF(Current position is not equal to Destination)
    MOVE(GET_DIRECTION() * GET_SPEED());
```

Second, the detection component has maximum, minimum, and normal detection distance as the property of the component. The component modifies the detection distance of the parent entity state variables as a result of the behavior. The modified detection distance is including environmental effects, and enemy entities that are existed in the perimeter of the detection distance are input to the detected enemy entities of the parent entity state variables by the simulation system. Following pseudo code is the representation of the modification of the detection distance by the entity state.

IF(*Destination has been updated*)

Detection Range = Normal Range

ELSE_IF(*Target has been updated*)

Detection Range = Minimum Range

ELSE

Detection Range = Maximum Range

Third, the fire control component decides that whether the entity is able to fire or not using the property of the component, such as a fire-able distance, a bullet counts, a loading time. If the fire control was ready to fire, it consumes the loading time and then fires a bullet to a target. When the component fires a bullet, it increases the noise emission level in the parent entity core part for a while. The fire message is sent to a bullet entity that is also designed as the core/shell. Following pseudo code represents the fire component can fire, when it fulfills both conditions of an entity state and component state.

IF(*Target has been updated &*

Fire system is enable to launch a weapon)

WAIT(*Preparation Time*);

LAUNCH_WEAPON();

Lastly, the protection component calculates damage amount using armor type and level in the property of the component, when the parent entity received impact. The calculated result is applied to the damaged level of the parent entity state variables. Following pseudo code represents the behavior of the protection component.

IF(*Impact has been updated*)

SET_DAMAGE(CAL_IMPACT());

In order to execute a simulation, simulation managers are composed of various managers to orchestrate the entire simulation system. There are six managers required to implement this proposed architecture; time manager, event scheduler, entity manager, scenario manager, detection manager, and damage manager. The entity manager manages core/shell based entities and sends events to the entities. The time manager makes the simulation clock proceed. The event scheduler indicates an event that is required to be occurred in the time which equals the current simulation time, and it reserves a next event. The entity manager checks all the entities when it received an event for checking the states of entities. The scenario manager modifies the destination and target entity of each entity by the updated states of the entities. The detection manager queries environmental data which are regarding to current positions of the entities, and it calculates detection probabilities among entities using the result of the query. The result of the calculation is reflected to the detected enemy entities of the state variable in the core part of each entity. The damage manager gives specific

amount of impact to an entity using the current position and property. When the damage of an entity is over its durability, the manager notices the entity manager that the entity is destroyed. The managers that require mathematical calculation model, such as a detection manager and a damage manager, contain engineering models inside, and it uses the model to obtain more sophisticated results during a simulation. Figure 3 describes the system design, and the system includes entities, managers, and synthetic battlefield as explained above. In order to construct the synthetic battlefield, the system contains the height-map for the terrain and the three-dimensional grid structure for the atmosphere.

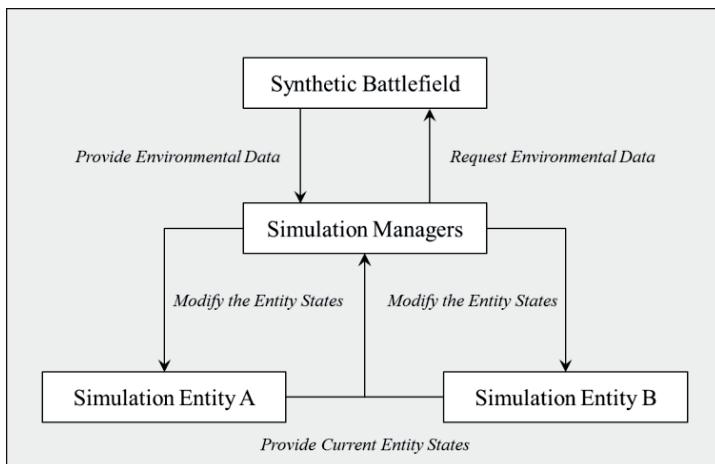


Figure 3. Engagement Simulation System Design

4. Implementation

The proposed architecture is implemented for an example, and the example system is described in this chapter. The example system generates a simulation entity based on the core part information inserted by user input. An agent component is created by input of information in the property of the component. The created agent component is connected to a shell part of an entity, and it allows the entity to operate a behavior of the component. The example system has terrain and precipitation for the synthetic battlefield, and it also comprises the simulation managers to manage entities and environmental data.

The example system is implemented by C++ programming language, and it is developed using MFC(Microsoft Foundation Class) library for UI(User-Interface) and OGRE3D (Object oriented Graphics Rendering Engine 3D) library for 3D visualization[17]. Figure 4 describes the generation of an entity by input for the property of the component in the example system.

The generated entity can perform various behaviors by connecting agent components to the shell part. Figure 5 represents the connection of the movement component in the example system. The entity that has the movement component is able to move on the terrain.

A cannon bullet that is launched by the fire control component is also designed as core/shell based. Figure 6 is UI for the generation of a cannon bullet that is created by

the user input for the property of the component. Figure 7 shows the process for the equipment of the created cannon bullet and launching the bullet to the enemy entity.



Figure 4. Entity Generation Process

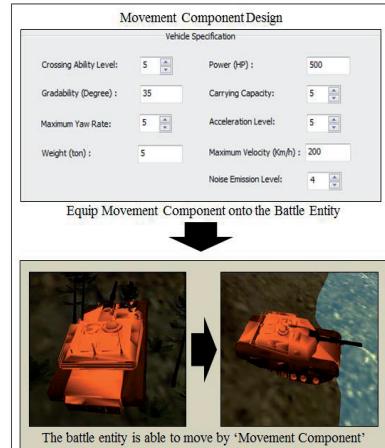


Figure 5. Equipment of Movement Component

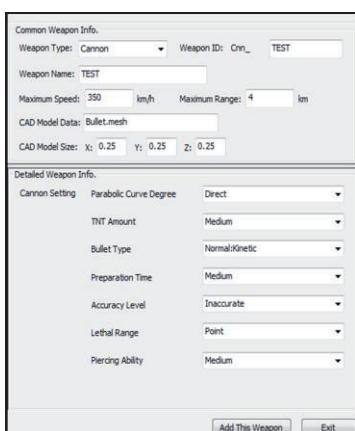


Figure 6. Core Part Design of a Bullet Entity

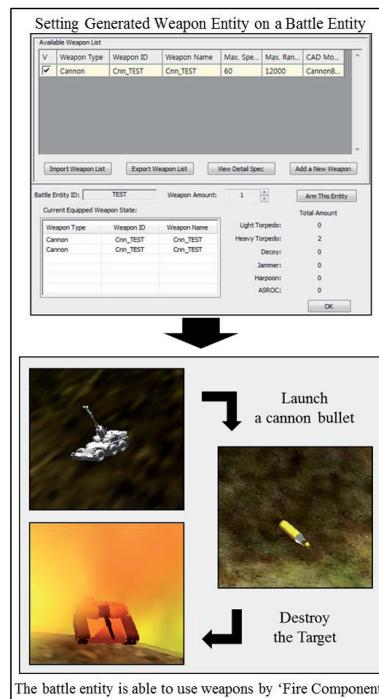


Figure 7. Equip and Launch a Bullet Entity

5. Concluding Remarks

In order to apply DM&S into practice, it is required the development framework of the simulation model which can design a simulation entity easily. Moreover, it is also required the execution of the engagement simulation on the synthetic battlefield which can represent the operational environment of the real weapon systems. Unfortunately, existing simulation tools are not enough to meet the requirements of the engagement simulation. The main objective of this paper is to propose an architecture that can fulfill the requirements.

The proposed architecture of this paper designs a simulation entity as the core/shell based, and the shell part has connections with agent components. An agent component refers the core part of its parent entity, and it performs own behavior based on the state variables of the core part. By the core/shell entity design methodology, a simulation entity is composable, reusable, and flexible. It is identified by the implementation of the example system of which is constructed based on the proposed architecture. The core/shell based simulation entity can reflect environmental effects to the behaviors by the simulation managers. Thus, the results of behaviors of simulation entities; attack, detection, and protection; in the example system are able to be changed by the environmental effects. In conclusion, the proposed architecture is effective to achieve the requirements, which is identified by the implementation of the example system.

Acknowledgements

This work was supported by the Defense Acquisition Program Administration (DAPA), the Agency for Defense Development (ADD), and Korea Association of Industry Academy and Research Institute (C00035790100384919) under the Contract No. UD110006MD (DAPA), UD100009DD & UD120035JD (ADD). The authors wish to express sincere gratitude for the financial support.

References

- [1] P. F. Gorman, The Future of Tactical Engagement Simulation. Proceedings of the 1991 Summer Computer Simulation Conference (1991), 1-12.
- [2] MSCO. 2012. Description of M&SCO. URL: <http://www.msco.mil/descMSCO.html>.
- [3] R. R. Hill, G. A. McIntyre, Applications of Discrete Event Simulation Modeling To Military Problems. Proceedings of the 2001 winter simulation conference (2001).
- [4] A. Maria, Introduction to Modeling and Simulation. Proceedings of the 1997 Winter Simulation Conference (1997), 7-13.
- [5] E. H. Page, R. Smith, Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists. Proceedings of the 1998 Winter Simulation Conference (1998), 53-60.
- [6] J. F. Keane, R. R. Lutz, S. E. Myers, J. E. Coolahan, An Architecture for Simulation Based Acquisition. Johns Hopkins APL Technical Digest, 21(3) (2000), 348-358.
- [7] S. C. Park, Y. Kwon, K. Seong, J. Pyun, Simulation Framework for Small Scale Engagement. Computer&Industrial Engineering 59 (2010), 463-472.
- [8] R. D. Smith, Essential Techniques for Military Modeling & Simulation. Proceedings of the 1998 winter simulation conference (1998).
- [9] A. Ilachinski, Artificial War: Multiagent-based Simulation of Combat. World Scientific Publishing Co. Pte. Ltd., 2004.
- [10] P. Phongsak, UTSAF: A Multi-Agent-Based Software Bridge for Interoperability between Distributed Military and Commercial Gaming Simulation. SIMULATION 80(12) (2004), 647-657.
- [11] M. C. Thomas, W. L. Thomas, M. S. Susan, Military applications of agent-based simulations. Proceedings of the 2004 winter simulation conference (2004).

- [12] CIL Ibrahim, MALA Murat, MABSIM: A Multi Agent Based Simulation Model of Military Unit Combat. Proc. of IEEE ICADIWT2009 (2009), 731-736.
- [13] J. Logsdon, D. Nash, M. Barnes, One Semi-Automated Forces(OneSAF): Capabilities, Architecture, and Processes. DMSC 2008 (2008).
- [14] A. Yang, A. Abbass, R. Sarker, WISDOM-II: A Network Centric Model for Warfare. LNAI 3683 (2005), 813-819.
- [15] S. C. Park, A Methodology for Creating a Virtual Model for a Flexible Manufacturing System. Computers in Industry 56 (2005), 734-746.
- [16] W. K. Hwam, Y. Chung, Y. Kwon, S. C. Park, Conversion the time dependent grid data of NetCDF to SEDRIS Transmittal Format. Proceedings of the 5th International Conference on IT & Multimedia (2011).
- [17] OGRE3D. 2012. Introduction of OGRE3D. URL: <http://www.ogre3d.org>.

Computational Intelligence for Business Collaboration (CIBC 2013)

Relationships Among the Concepts of Reduct in Incomplete Decision Tables

Nguyen Long GIANG ^{a,1}, Vu Van DINH ^b

^a*Institute of Information Technology, VAST, Viet Nam*

^b*Electric Power University, Ha Noi, Viet Nam*

Abstract. With regard to attribute reduction research directions in tolerance rough set, this paper studies the relationships among concepts of reduct of incomplete decision tables. These results allows us to investigate the nature of concepts of reduct and also give us the theoretical basis for evaluation of attribute reduction methods in order to find new methods more effectively.

Keywords. Rough set theory, incomplete decision table, tolerance rough set, attribute reduction, reduct

1. Introduction

Classical rough set theory based on equivalent relation has been introduced by Pawlak [1] as one of the effective tools for rule induction, object classification in complete decision tables. Attribute reduction, one of the crucial problems in classical rough set theory, has attracted the attention of many researchers in recent years. In fact, there are many cases that decision tables contain missing values for at least one conditional attribute in the value set of that attribute and these decision tables are called incomplete decision tables. To obtain decision rules directly from incomplete decision tables, Marzena Kryszkiewicz [2] has defined a tolerance relation based on the equivalent relation in classical rough set and proposed tolerance rough set. Using this tolerance rough set, many researchers have proposed different concepts of reduct based on different measures and proposed attribute reduction methods in incomplete decision tables. Marzena Kryszkiewicz [2] introduced the concept of *reduct based on generalized decision*. Zuqiang Meng et al [3] proposed the concept of *reduct based on positive region*. Huang B et al [4] proposed the concept of *reduct based on information quantity*. Nguyen Long Giang et al [5] proposed the concept of *reduct based on metric*. Zhou X.Z et al, Renpu Li et al [6,7] proposed *distribution reduct, assignment reduct*. According to the discernibility matrix approach in traditional rough set theory [8], Huasheng Zou et al [9] proposed the concept of *reduct based on discernibility matrix*.

It has been shown that a reduct is the result of an attribute reduction method. Thus, finding the relationships among reducts is very necessary for comparison and evaluation of attribute reduction methods. In complete decision tables, Wei Wei et al [10] reviewed

¹Corresponding Author: Nguyen Long Giang, Institute of Information Technology, Vietnamese Academy of Science and Technology (VAST), Viet Nam; E-mail: nlgiang@ioit.ac.vn

and investigated fully the relationship among reducts. However, in incomplete decision tables, the results of research on the relationships among reducts are limited. Some typical results related to this research direction are: Renpu Li et al, Zuqiang Meng et al [3,6] showed that reduct based on positive region, reduct based on generalized decision, distribution reduct and assignment reduct are the same for consistent incomplete decision tables. For inconsistent incomplete decision tables, Renpu Li et al [6] proved that reduct based on generalized function is equivalent to assignment reduct. Huasheng Zou et al [9] proved that reduct based on generalized decision is equivalent to reduct based on discernibility matrix.

In this paper, we study fully the relationship among the above reducts in incomplete decision tables. As a result, reducts are grouped according to this relationship. The structure of this paper is as follows. Section 2 presents some basic concepts in tolerance rough set and some concepts of reduct in incomplete decision tables. Section 3 presents the relationships among reducts. The conclusion and future research are presented in the last section.

2. Some Concepts About Tolerance Rough Set and Reducts

In this section, we present some basic concepts about tolerance rough set which proposed by Marzena Kryszkiewicz [2] and some concepts about reducts of incomplete decision tables.

An information system is a pair $IS = (U, A)$, where the set U denotes the *universe of objects* and A is the set of *attributes*, i.e. mappings of the form: $a : U \rightarrow V_a$. V_a is called the *value set* of attribute a . If V_a contains a missing value for at least one attribute $a \in A$, then IS is called an *incomplete information system*, otherwise it is *complete*. Further on, we will denote the missing value by *. An incomplete decision table IDS is an incomplete information system where $d, d \notin A$ and $* \notin V_d$, is a distinguished attribute called *decision attribute*, and the elements of A are called *conditional attributes*.

Let $IIS = (U, A)$ be an incomplete information system. For any attribute set $P \subseteq A$. We define a binary relation on U as follows:

$$SIM(P) = \{(u, v) \in U \times U \mid \forall a \in P, f(u, a) = f(v, a) = *' \vee f(u, a) = *' \vee f(v, a) = *'\}$$

$SIM(P)$ is a tolerance relation on U . It can be easily shown that $SIM(P) = \cap_{a \in P} SIM(\{a\})$. Let $U/SIM(P)$ denote the family sets $\{S_P(u) \mid u \in U\}$ where $S_P(u) = \{v \in U \mid (u, v) \in SIM(P)\}$ is the maximal set of objects which are possibly indistinguishable by P with u . A member $S_P(u)$ in $U/SIM(P)$ is called a tolerance class or a granule of information. It is clear that the tolerance classes in $U/SIM(P)$ do not constitute a partition of U in general. They constitute a covering of U , i.e., $S_P(u) \neq \emptyset$ for every $u \in U$, and $\cup_{u \in U} S_P(u) = U$.

For any $B \subseteq A$, $X \subseteq U$, B -lower approximation of X is the set $\underline{BX} = \{u \in U \mid S_B(u) \subseteq X\} = \{u \in X \mid S_B(u) \subseteq X\}$, B -upper approximation of X is the set $\overline{BX} = \{u \in U \mid S_B(u) \cap X \neq \emptyset\} = \cup \{S_B(u) \mid u \in U\}$, B -boundary region of X is the set $BN_P(X) = \overline{PX} - \underline{PX}$. For such approximation set, B -positive region with respect to d is defined as

$$POS_B(\{d\}) = \bigcup_{X \in U/\{d\}} (\underline{BX})$$

Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table, for any $B \subseteq A$ and $u \in U$, $\partial_B(u) = \{f_d(v) | v \in S_B(u)\}$ will be called *generalized decision* in IDS . If $|\partial_C(u)| = 1$ for any $u \in U$ then IDS is *consistent*, otherwise it is *inconsistent*. According to the definition of positive region, IDS is consistent if and only if $POS_A(\{d\}) = U$, otherwise it is inconsistent.

In next content, we present some concepts about reducts of incomplete decision tables which is related to this paper.

Kryszkiewicz [2] present the first definition of reduct based on generalized decision.

Definition 1. [2] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If $R \subseteq A$ satisfies

- (1) $\partial_R(u) = \partial_A(u)$ for any $u \in U$
- (2) $\forall r \in R, R^1 = R - \{r\}$ is not satisfied (1)

then R is called a *reduct of IDS based on generalized decision*.

Zuqiang Meng et al [3] propose the concept of reduct based on positive region.

Definition 2. [3] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If $R \subseteq A$ satisfies

- (1) $POS_R(\{d\}) = POS_A(\{d\})$
- (2) $\forall R^1 \subset R, POS_{R^1}(\{d\}) \neq POS_A(\{d\})$

then R is called a *reduct of IDS based on positive region*.

Huang B et al [4] propose the concept of reduct based on information quantity. For any $B \subseteq A$, the information quantity of B with respect to d is $I(B|\{d\}) = I(B \cup \{d\}) - I(B)$ where $I(B) = 1 - \frac{1}{|U|^2} \sum_{i=1}^n |S_B(u_i)|$ and $U = \{u_1, u_2, \dots, u_n\}$.

Definition 3. [4] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If $R \subseteq A$ satisfies

- (1) $I(R|\{d\}) = I(A|\{d\})$
- (2) $\forall R^1 \in R, I(R^1|\{d\}) \neq I(A|\{d\})$

then R is called a *reduct of IDS based on information quantity*.

Based on metric, Nguyen Long Giang et al [5] define a reduct of an incomplete decision table and propose an attribute reduction algorithm. For $P \subseteq A$, suppose that we have two coverings $K(P) = U/SIM(P) = \{S_P(u_1), S_P(u_2), \dots, S_P(u_{|U|})\}$, $K(A) = U/SIM(A) = \{S_A(u_1), S_A(u_2), \dots, S_A(u_{|U|})\}$. Then, the metric between P and A is defined as

$$d_E(K(P), K(A)) = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{|S_P(u_i)| - |S_A(u_i)|}{|U|}$$

Definition 4. [5] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If $R \subseteq A$ satisfies

- (1) $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$
- (2) $\forall R^1 \subseteq R, d_E(K(R^1), K(R^1 \cup \{d\})) \neq d_E(K(A), K(A \cup \{d\}))$

then R is called a *reduct of IDS based on metric*.

Huasheng ZOU et al [9] propose the concept of reduct based on discernibility matrix. The discernibility matrix of IDS is $M = [m_{ij}]_{n \times n}$, elements m_{ij} are defined as

$$m_{ij} = \begin{cases} \{a | a \in A, a(u_i) \neq * \wedge a(u_j) \neq * \wedge a(u_i) \neq a(u_j)\} & d(u_j) \notin \partial_A(u_i) \\ \emptyset & d(u_j) \in \partial_A(u_i) \end{cases}$$

Definition 5. [9] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and the discernibility matrix $M = [m_{ij}]_{n \times n}$. If $R \subseteq A$ satisfies

Table 1. Reducts in incomplete decision tables

Notation	Description
R_P	Reduct based on positive region
R_∂	Reduct based on generalized decision
R_δ	Assignment reduct
R_M	Reduct based on discernibility matrix
R_D	Reduct based on metric
R_I	Reduct based on information quantity
R_{TM}	Reduct based on tolerance matrix
R_μ	Distribution reduct

(1) $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$

(2) For any $r \in R$, $R - \{r\}$ is not satisfied (1)

then R is called a *reduct of IDS based on discernibility matrix*.

According to this approach, we propose the concept of reduct based on tolerance matrix. The tolerance matrix of *IDS* is $TM = [m_{ij}]_{n \times n}$, elements m_{ij} are defined as:

$$m_{ij} = \begin{cases} \{a \mid a \in A, a(u_i) \neq * \wedge a(u_j) \neq * \wedge a(u_i) \neq a(u_j)\} & d(u_i) \neq d(u_j) \\ \emptyset & d(u_i) = d(u_j) \end{cases}$$

Definition 6. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and the tolerance matrix $TM = [m_{ij}]_{n \times n}$. If $R \subseteq A$ satisfies

(1) $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$

(2) For any $r \in R$, $R - \{r\}$ is not satisfied (1)

then R is called a *reduct of IDS based on tolerance matrix*.

In addition, in the papers [6,7] the authors introduce the concept of distribution reduct, assignment reduct.

Definition 7 [6,7] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table, $R \subseteq A$, $U = \{u_1, \dots, u_{|U|}\}$, $U \setminus \{d\} = \{Y_1, \dots, Y_m\}$. For any $u_i \in U$, we set

$$\mu_R(u_i) = (Y_1^R(u_i), \dots, Y_m^R(u_i)) \text{ where } Y_j^R(u_i) = \frac{|Y_j \cap S_R(u_i)|}{|S_R(u_i)|}, j = 1, \dots, m.$$

$$\delta_R(u_i) = \{Y_j : Y_j \cap S_R(u_i) \neq \emptyset\}$$

(1) R is called a *distribution reduct of IDS* iff $\mu_R(u_i) = \mu_A(u_i)$ for $i = 1, \dots, |U|$ and for $\forall P^1 \subset P$, there exists $u_j \in U$ such that $\mu_R(u_j) \neq \mu_A(u_j)$.

(2) R is called a *assignment reduct of IDS* iff $\delta_R(u_i) = \delta_A(u_i)$ for $i = 1, \dots, |U|$ and for $\forall P^1 \subset P$, there exists $u_j \in U$ such that $\delta_R(u_j) \neq \delta_A(u_j)$.

3. The Relationships Among Reducts

In this section, we summarize and study the relationships among the concepts of reduct which was presented in section 2 of consistent and inconsistent incomplete decision tables. To describe briefly, we denote reducts as **Table 1**.

Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If *IDS* is consistent, it is known from [3,6] that $R_P, R_\partial, R_\mu, R_\delta$ are the same. If *IDS* is inconsistent, Renpu Li et. al. [6] prove that R_∂ is equivalent to R_δ , Huasheng ZOU et. al. [9] prove that R_∂ is equivalent to R_M . Consequently, If *IDS* is inconsistent then the reducts $R_\partial, R_\delta, R_M$ are the same.

In next content, we continue to study the relationships among the above reducts.

3.1. Relationships Among R_D , R_I , R_{TM}

Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. In this subsection, we prove that R_D , R_I and R_{TM} are the same in both cases: IDS is consistent and inconsistent.

Proposition 1. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and $R \subseteq A$. Then, $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ if and only if $I(R| \{d\}) = I(A| \{d\})$.

Proof. Let us consider the incomplete decision table $IDS = (U, A \cup \{d\})$ where $U = \{u_1, u_2, \dots, u_n\}$ and $B \subseteq A$. Since $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$, according to the definition of metric, we have:

$$\begin{aligned} & \frac{1}{|U|} \sum_{i=1}^{|U|} \left(\frac{|S_R(u_i)| - |S_{R \cup \{d\}}(u_i)|}{|U|} \right) = \frac{1}{|U|} \sum_{i=1}^{|U|} \left(\frac{|S_A(u_i)| - |S_{A \cup \{d\}}(u_i)|}{|U|} \right) \\ & \Leftrightarrow \left(1 - \frac{1}{|U|^2} \sum_{i=1}^{|U|} |S_{R \cup \{d\}}(u_i)| \right) - \left(1 - \frac{1}{|U|^2} \sum_{i=1}^{|U|} |S_R(u_i)| \right) \\ & = \left(1 - \frac{1}{|U|^2} \sum_{i=1}^{|U|} |S_{A \cup \{d\}}(u_i)| \right) - \left(1 - \frac{1}{|U|^2} \sum_{i=1}^{|U|} |S_A(u_i)| \right) \end{aligned}$$

According to the definition of information quantity, we have:

$$I(R \cup \{d\}) - I(R) = I(A \cup \{d\}) - I(A) \Leftrightarrow I(R| \{d\}) = I(A| \{d\})$$

We can conclude from Proposition 1 that reduct based on metric is equivalent to reduct based on information quantity, or R_D is equivalent to R_I .

Proposition 2. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table, $R \subseteq A$ and $TM = [m_{ij}]_{n \times n}$ is the tolerance matrix of IDS . Then, $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ if and only if $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$.

Proof. Let us consider the incomplete decision table $IDS = (U, A \cup \{d\})$ where $U = \{u_1, u_2, \dots, u_n\}$ and $R \subseteq A$. Since $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$, according to the definition of metric, we have:

$$\begin{aligned} & \frac{1}{|U|} \sum_{i=1}^{|U|} \left(\frac{|S_R(u_i)| - |S_{R \cup \{d\}}(u_i)|}{|U|} \right) = \frac{1}{|U|} \sum_{i=1}^{|U|} \left(\frac{|S_A(u_i)| - |S_{A \cup \{d\}}(u_i)|}{|U|} \right) \\ & \Leftrightarrow |S_R(u_i)| - |S_{R \cup \{d\}}(u_i)| = |S_A(u_i)| - |S_{A \cup \{d\}}(u_i)| \text{ for any } u_i \in U \quad (1) \end{aligned}$$

$$\text{It is clear that } S_{R \cup \{d\}}(u_i) \subseteq S_R(u_i), S_{A \cup \{d\}}(u_i) \subseteq S_A(u_i), \text{ so Eq. (1) is equivalent to } |S_R(u_i) - S_{R \cup \{d\}}(u_i)| = |S_A(u_i) - S_{A \cup \{d\}}(u_i)| \text{ for any } u_i \in U \quad (2)$$

$$\text{Since } S_A(u_i) \subseteq S_R(u_i) \Leftrightarrow S_A(u_i) - S_{\{d\}}(u_i) \subseteq S_R(u_i) - S_{\{d\}}(u_i)$$

$$\Leftrightarrow S_A(u_i) - S_A(u_i) \cap S_{\{d\}}(u_i) \subseteq S_R(u_i) - S_R(u_i) \cap S_{\{d\}}(u_i)$$

$$\Leftrightarrow S_A(u_i) - S_{A \cup \{d\}}(u_i) \subseteq S_R(u_i) - S_{R \cup \{d\}}(u_i)$$

So Eq. (2) is equivalent to:

$$S_R(u_i) - S_{R \cup \{d\}}(u_i) = S_A(u_i) - S_{A \cup \{d\}}(u_i) \text{ for any } u_i \in U \quad (3)$$

i) We prove that if Eq. (3) is satisfied then $R \cap m_{ij} \neq \emptyset$ for any $u_i \in U$.

Suppose that there exists $m_{i_0 j_0} \neq \emptyset$ such that $R \cap m_{i_0 j_0} = \emptyset$. Then there exists $u_{i_0}, u_{j_0} \in U$ such that $d(u_{i_0}) \neq d(u_{j_0})$. u_{i_0}, u_{j_0} are indistinguishable on R and u_{i_0}, u_{j_0} are distinguishable on $A-R$, that is $u_{j_0} \notin S_A(u_{i_0})$ and $u_{j_0} \in S_R(u_{i_0})$.

Since $u_{j_0} \notin S_A(u_{i_0})$, we have $u_{j_0} \notin S_A(u_{i_0}) - S_{A \cup \{d\}}(u_{i_0})$ (*)

Since $u_{j_0} \in S_R(u_{i_0})$ and $d(u_{i_0}) \neq d(u_{j_0})$, we have $u_{j_0} \in S_R(u_{i_0}) - S_R(u_{i_0}) \cap S_{\{d\}}(u_{i_0})$ or $u_{j_0} \in S_R(u_{i_0}) - S_{R \cup \{d\}}(u_{i_0})$. (**)

From (*) and (**), it follows that $S_R(u_i) - S_{R \cup \{d\}}(u_i) \neq S_A(u_i) - S_{A \cup \{d\}}(u_i)$, which contradicts to Eq. (3). Consequently, we can conclude that if Eq. (3) is satisfied then $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$.

Table 2. The incomplete decision table of Example 1

U	a_1	a_2	a_3	d
u_1	1	1	0	0
u_2	1	1	0	1
u_3	1	*	0	0
u_4	*	2	2	2
u_5	2	2	*	0

ii) Conversely, we need to prove that if $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$ then Eq. (3) is satisfied.

Suppose that there exists u_{i_0} such that $S_R(u_{i_0}) - S_{R \cup \{d\}}(u_{i_0}) \neq S_A(u_{i_0}) - S_{A \cup \{d\}}(u_{i_0})$. Because of $S_A(u_{i_0}) - S_{A \cup \{d\}}(u_{i_0}) \subseteq S_R(u_{i_0}) - S_{R \cup \{d\}}(u_{i_0})$, there exists $u_{j_0} \in U$ such that $u_{j_0} \in S_R(u_{i_0}) - S_{R \cup \{d\}}(u_{i_0})$ and $u_{j_0} \notin S_A(u_{i_0}) - S_{A \cup \{d\}}(u_{i_0})$. Since $u_{j_0} \in S_R(u_{i_0}) - S_{R \cup \{d\}}(u_{i_0})$, we have $d(u_{j_0}) \neq d(u_{i_0})$, it follows from $u_{j_0} \notin S_A(u_{i_0}) - S_{A \cup \{d\}}(u_{i_0})$ that $u_{j_0} \notin S_A(u_{i_0})$. According to the definition of tolerance matrix, there exists $m_{i_0 j_0} \neq \emptyset$ such that for any $a \in m_{i_0 j_0}$ we have $a \notin R$ (because $u_{j_0} \in S_R(u_{i_0})$), i.e. $R \cap m_{i_0 j_0} = \emptyset$, which contradicts to the precondition $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$. Consequently, we can conclude that if $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$ then Eq. (3) is satisfied.

From i) and ii) we can conclude that $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ if and only if $R \cap m_{ij} \neq \emptyset$ for any $m_{ij} \neq \emptyset$.

It is known from Proposition 2 that the concept of reduct based on metric is equivalent to the concept of reduct based on tolerance matrix, or R_D is equivalent to R_{TM} . It follows from Proposition 1 that R_D , R_I and R_{TM} are the same.

3.2. Relationship Between R_∂ and R_P

Proposition 3. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and $R \subseteq A$. If $\partial_R(u) = \partial_A(u)$ for any $u \in U$ then $POS_R(\{d\}) = POS_A(\{d\})$.

Proof. Suppose that $POS_R(\{d\}) \neq POS_A(\{d\})$, then there exists $u_0 \in U$ such that $u_0 \in POS_A(\{d\})$ and $u_0 \notin POS_R(\{d\})$. It follows from $u_0 \in POS_A(\{d\})$ that $|\partial_A(u_0)| = 1$, it follows from $u_0 \notin POS_R(\{d\})$ that $|\partial_R(u_0)| \neq 1$. Hence, $|\partial_R(u_0)| \neq |\partial_A(u_0)|$. Since $S_A(u_0) \subseteq S_R(u_0)$ we have $\partial_A(u_0) \subseteq \partial_R(u_0)$, which combines with $|\partial_R(u_0)| \neq |\partial_A(u_0)|$, we obtain $\partial_A(u_0) \neq \partial_R(u_0)$, which contradicts to the precondition $\partial_R(u) = \partial_A(u)$ for any $u \in U$. Consequently, we can conclude that if $\partial_R(u) = \partial_A(u)$ for any $u \in U$ then $POS_R(\{d\}) = POS_A(\{d\})$.

Remark. If IDS is inconsistent then the inverse of Proposition 3 does not hold. This is illustrated by the following Example 1

Example 1. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table where $U = \{u_1, u_2, u_3, u_4, u_5\}$, $A = \{a_1, a_2, a_3\}$ as **Table 2**.

We have that $\partial_A(u_1) = \partial_A(u_2) = \partial_A(u_3) = \{0, 1\}$, $\partial_A(u_4) = \partial_A(u_5) = \{0, 2\}$. Hence, IDS is inconsistent and it is easy to see that $POS_A(\{d\}) = \emptyset$. Let us consider $R = \{a_1, a_2\}$, it is easy to see that $POS_R(\{d\}) = POS_A(\{d\}) = \emptyset$. However, $\partial_A(u_3) = \{u_1, u_2, u_3\}$ and $\partial_R(u_3) = \{u_1, u_2, u_3, u_4\}$, thus $\partial_R(u_3) \neq \partial_A(u_3)$.

It is known from Proposition 3 that if R_∂ is a reduct based on generalized decision then there exists a R_P such that $R_P \subseteq R_\partial$ where R_P is a reduct based on positive region.

If IDS is consistent then $POS_R(\{d\}) = POS_A(\{d\}) = U$, this means that for any $u \in U$ we have $|\partial_R(u)| = |\partial_A(u)| = 1$, or $\partial_R(u) = \partial_A(u)$. Consequently, $\partial_R(u) = \partial_A(u)$

Table 3. The incomplete decision table of Example 2

U	a_1	a_2	a_3	d
u_1	1	*	0	1
u_2	1	1	0	0
u_3	1	*	0	0
u_4	*	2	2	0
u_5	2	2	*	1

for any $u \in U$ if and only if $POS_R(\{d\}) = POS_A(\{d\})$, this means that R_∂ is equivalent to R_P .

3.3. Relationship Between R_D and R_∂

Proposition 4. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and $R \subseteq A$. If $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ then $\forall u \in U$, $\partial_R(u) = \partial_A(u)$

Proof. Let us consider the incomplete decision table $IDS = (U, A \cup \{d\})$ where $U = \{u_1, u_2, \dots, u_n\}$ and $R \subseteq A$. Since $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$, according to the proof of Proposition 2, Eq. (3), we have:

$$S_R(u_i) - S_{R \cup \{d\}}(u_i) = S_A(u_i) - S_{A \cup \{d\}}(u_i) \text{ for any } u_i \in U \quad (4)$$

Otherwise

$$S_B(u_i) = (S_B(u_i) \cap S_{\{d\}}(u_i)) \cup (S_B(u_i) - (S_B(u_i) \cap S_{\{d\}}(u_i)))$$

$$S_C(u_i) = (S_C(u_i) \cap S_{\{d\}}(u_i)) \cup (S_C(u_i) - (S_C(u_i) \cap S_{\{d\}}(u_i)))$$

$$\text{Suppose that } d_i = d(u_i), B_i = \{d(v_i) | v_i \in S_B(u_i) - (S_B(u_i) \cap S_{\{d\}}(u_i))\},$$

$$C_i = \{d(v_i) | v_i \in S_C(u_i) - (S_C(u_i) \cap S_{\{d\}}(u_i))\}$$

Then we have that

$$\partial_B(u_i) = \{d(v_i) | v_i \in (S_B(u_i) \cap S_{\{d\}}(u_i)) \cup (S_B(u_i) - (S_B(u_i) \cap S_{\{d\}}(u_i)))\} = \{d_i\} \cup B_i$$

$$\partial_C(u_i) = \{d(v_i) | v_i \in (S_C(u_i) \cap S_{\{d\}}(u_i)) \cup (S_C(u_i) - (S_C(u_i) \cap S_{\{d\}}(u_i)))\} = \{d_i\} \cup C_i$$

According to Eq. (4), we obtain $B_i = C_i$, thus $\partial_B(u_i) = \partial_C(u_i)$ for any $u \in U$.

Remark. If IDS is inconsistent then the inverse of Proposition 4 does not hold because the condition $\forall u_i \in U, \partial_B(u_i) = \partial_C(u_i)$ only preserves the generalized decision $\partial_B(u_i)$ of the tolerance class $S_B(u_i)$, also the condition $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ preserves inconsistent objects with respect to u_i of the tolerance class $S_B(u_i)$ (this condition is more closer). This is illustrated by the following Example 2.

Example 2. Let us consider the incomplete decision table $IDS = (U, A \cup \{d\})$ where $U = \{u_1, u_2, u_3, u_4, u_5\}$, $A = \{a_1, a_2, a_3\}$ as **Table 3**.

We have that $S_A(u_1) = S_A(u_2) = S_A(u_3) = \{u_1, u_2, u_3\}$, $S_A(u_4) = S_A(u_5) = \{u_4, u_5\}$, $\partial_A(u_1) = \partial_A(u_2) = \partial_A(u_3) = \partial_A(u_4) = \partial_A(u_5) = \{0, 1\}$. So IDS is inconsistent. Let us consider $R = \{a_1, a_2\}$, we have that $S_R(u_1) = S_R(u_3) = \{u_1, u_2, u_3, u_4\}$, $S_R(u_2) = \{u_1, u_2, u_3\}$, $S_R(u_4) = \{u_1, u_3, u_4, u_5\}$, $S_R(u_5) = \{u_4, u_5\}$, $\partial_R(u_1) = \partial_R(u_2) = \partial_R(u_3) = \partial_R(u_4) = \partial_R(u_5) = \{0, 1\}$. Thus, $\forall u_i \in U, i = 1..5, \partial_R(u_i) = \partial_A(u_i)$. Otherwise

$$d_E(K(A), K(A \cup \{d\})) = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{|S_A(u_i)| - |S_{A \cup \{d\}}(u_i)|}{|U|} = \frac{1}{4.4} (2 + 1 + 1 + 1 + 1) = \frac{6}{16}$$

$$d_E(K(R), K(R \cup \{d\})) = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{|S_R(u_i)| - |S_{R \cup \{d\}}(u_i)|}{|U|} = \frac{1}{4.4} (3 + 1 + 1 + 2 + 1) = \frac{8}{16}$$

Consequently, $d_E(K(R), K(R \cup \{d\})) \neq d_E(K(A), K(A \cup \{d\}))$.

It is known from Proposition 4 that if R_D is a reduct based on metric then there exists a R_∂ such that $R_\partial \subseteq R_D$ where R_∂ is a reduct based on generalized decision.

If IDS is consistent, it follows from the condition $\forall u_i \in U, |\partial_R(u_i)| = |\partial_A(u_i)| = 1$ that $S_R(u_i) = S_{R \cup \{d\}}(u_i)$ and $S_A(u_i) = S_{A \cup \{d\}}(u_i)$ for any $u_i \in U$. According to the definition of metric, we have

$$d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\})) = 0$$

Consequently, $d_E(K(R), K(R \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ if and only if $\forall u_i \in U, \partial_R(u_i) = \partial_A(u_i)$, this means that R_D is equivalent to R_∂ .

3.4. Relationship Between R_μ and R_∂

Proposition 5. Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table and $R \subseteq A$. If $\forall u \in U, \mu_R(u) = \mu_A(u)$ then $\forall u \in U, \partial_R(u) = \partial_A(u)$.

Proof. Let us consider the incomplete decision table $IDS = (U, A \cup \{d\})$ where $U = \{u_1, u_2, \dots, u_n\}$ and $R \subseteq A$. Suppose that there exists $u_0 \in U$ such that $\partial_R(u_0) \neq \partial_A(u_0)$. Then there exists $d_0 \in \partial_R(u_0)$ such that $d_0 \notin \partial_A(u_0)$, where $d_0 \in Y_0, Y_0 \in U/\{d\}$. Since $d_0 \notin \partial_A(u_0)$ we have $Y_0 \cap S_A(u_0) = \emptyset$. Otherwise, since $d_0 \in \partial_R(u_0)$ we have $Y_0 \cap S_R(u_0) \neq \emptyset$. Thus, we have $Y_0 \cap S_A(u_0) \neq Y_0 \cap S_R(u_0)$, that is

$$\frac{|Y_0 \cap S_R(u_0)|}{|S_R(u_0)|} \neq \frac{|Y_0 \cap S_A(u_0)|}{|S_A(u_0)|},$$

according to the definition of distribution function $\mu_R(u_i)$ (Definition 7), we have $Y_0^R(u_0) \neq Y_0^A(u_0)$, so $\mu_R(u_0) \neq \mu_A(u_0)$, which contradicts to the precondition $\forall u \in U, \mu_R(u) = \mu_A(u)$. Consequently, we can conclude that if $\forall u \in U, \mu_R(u) = \mu_A(u)$ then $\forall u \in U, \partial_R(u) = \partial_A(u)$.

Remark. If IDS is inconsistent then the inverse of Proposition 5 does not hold because the condition $\forall u_i \in U, \partial_B(u_i) = \partial_C(u_i)$ only preserves the generalized decision $\partial_B(u_i)$ of the tolerance class $S_B(u_i)$, also the condition $\forall u_i \in U, \mu_R(u_i) = \mu_A(u_i)$ preserves the distribution function of u_i (this condition is more closer). This is illustrated by the following Example 3.

Example 3. (continued from Example 2) We have $\forall u_i \in U, i = 1..5, \partial_R(u_i) = \partial_A(u_i)$. Suppose that $U/\{d\} = \{Y_1, Y_2\}$ where $Y_1 = \{u_1, u_5\}, Y_2 = \{u_2, u_3, u_4\}$. Let us consider the object $u_1 \in U$.

$$\mu_A(u_1) = (Y_1^A(u_1), Y_2^A(u_1)) = \left(\frac{1}{3}, \frac{2}{3}\right), \mu_R(u_1) = (Y_1^R(u_1), Y_2^R(u_1)) = \left(\frac{1}{4}, \frac{3}{4}\right)$$

Thus, $\mu_R(u_1) \neq \mu_A(u_1)$.

It is known from Proposition 5 that if R_μ is a distribution reduct then there exists a R_∂ such that $R_\partial \subseteq R_\mu$ where R_∂ is a reduct based on generalized decision.

If IDS is consistent then $\forall u_i \in U, |\partial_B(u_i)| = |\partial_C(u_i)| = 1$. Suppose that $d_i = d(u_i)$ where $d_i \in Y_i, Y_i \in U/\{d\}$. Then,

$$Y_i^R(u_i) = \frac{|Y_i \cap S_R(u_i)|}{|S_R(u_i)|} = \frac{|S_R(u_i)|}{|S_R(u_i)|} = 1 \text{ and } Y_i^A(u_i) = \frac{|Y_i \cap S_A(u_i)|}{|S_A(u_i)|} = \frac{|S_A(u_i)|}{|S_A(u_i)|} = 1 \quad (5)$$

For $j \neq i, j = 1, \dots, m$, the elements of distribution function $\mu_R(u_i), \mu_A(u_i)$ of u_i as follows

$$Y_j^R(u_i) = \frac{|Y_j \cap S_R(u_i)|}{|S_R(u_i)|} = \frac{|\emptyset|}{|S_R(u_i)|} = 0 \text{ and } Y_j^A(u_i) = \frac{|Y_j \cap S_A(u_i)|}{|S_A(u_i)|} = \frac{|\emptyset|}{|S_A(u_i)|} = 0 \quad (6)$$

According to Eq. (5), Eq. (6) we have $\mu_R(u_i) = \mu_A(u_i)$ for any $u_i \in U$. Consequently, we can conclude that $\mu_R(u_i) = \mu_A(u_i)$ if and only if $\partial_R(u) = \partial_A(u)$ for any $u \in U$, this means that R_μ is equivalent to R_∂ .

3.5. Summary of the Relationships Among the Concepts of Reduct

From the published research and the mentioned above results, in this subsection we summarize the relationships among the concepts of reduct of incomplete decision table.

Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If IDS is consistent, the reducts $R_P, R_\partial, R_\delta, R_M, R_D, R_I, R_{TM}, R_\mu$ are the same. If IDS is inconsistent, the reducts $R_\partial, R_\delta, R_{TM}$ are the same, the reducts R_D, R_I, R_{TM} are the same. For convenience, the reducts $R_\partial, R_\delta, R_{TM}$ are denoted by R_I , the reducts R_D, R_I, R_{TM} are denoted by R_{II} . Hence, IDS have four kinds of reducts: R_P, R_I, R_{II}, R_μ . Suppose that $RED_P(A)$ is the set of all reducts R_P , $RED_I(A)$ is the set of all reducts R_I , $RED_{II}(A)$ is the set of all reducts R_{II} and $RED_\mu(A)$ is the set of all reducts R_μ . Then, relationships among kinds of reducts as follows

1. If R_{II} is a reduct in $RED_{II}(A)$ then there exists a reduct R_I in $RED_I(A)$ and a reduct R_P in $RED_P(A)$ such that $R_P \subseteq R_I \subseteq R_{II}$.
2. If R_μ is a reduct in $RED_\mu(A)$ then there exists a reduct R_I in $RED_I(A)$ and a reduct R_P in $RED_P(A)$ such that $R_P \subseteq R_I \subseteq R_\mu$.

From the above relationships, we have the following corollary.

Corollary 1. Let $IDS = (U, A \cup \{d\})$ be an inconsistent incomplete decision table, $RED_P(A), RED_I(A), RED_{II}(A)$ and $RED_\mu(A)$ are the set of all reducts R_P , reducts R_I , reducts R_{II} and reducts R_μ respectively. Then, we have:

- 1) $\min \{|R_P| : R_P \in RED_P(A)\} \leq \min \{|R_I| : R_I \in RED_I(A)\} \leq \min \{|R_{II}| : R_{II} \in RED_{II}(A)\}$
- 2) $\min \{|R_P| : R_P \in RED_P(A)\} \leq \min \{|R_I| : R_I \in RED_I(A)\} \leq \min \{|R_\mu| : R_\mu \in RED_\mu(A)\}$

4. Conclusion

In this paper, we have summarized and studied fully the relationship between eight concepts of reduct in incomplete decision tables. For consistent incomplete decision tables, the reducts are the same. For inconsistent incomplete decision tables, the reducts are divided into four kinds and the relationships among the kinds of reduct are investigated. This results are really significant, it is the theoretical basis to evaluate attribute reduction methods as well as develop new attribute reduction methods more effectively. Based on the results of this paper, our future research is to study the changes of evaluation measures of the performance of decision rules based on the concepts of reduct to solve fully the problems of evaluation of attribute reduction methods in incomplete decision tables.

References

- [1] Pawlak Z., Rough sets, *International Journal of Information and Computer Sciences*, 11(5) (1982), 341–356.
- [2] Kryszkiewicz M., Rough set approach to incomplete information systems, *Information Science*, Vol. 112 (1998), 39–49.
- [3] Meng Z. Q., Shi Z. Z., A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets, *Information Sciences*, Vol. 179 (2009), 2774–2793.
- [4] Huang B., Li H. X., Zhou X. Z., Attribute Reduction Based on Information Quantity under Incomplete Information Systems, *Systems Application Theory and Practice*, Vol. 34 (2005), 55–60.

- [5] Giang N. L, Tung N. T., Vu Duc Thi, A New Method for Attribute Reduction in Incomplete Decision Tables Based on Metric, *Journal of Computer Science and Cybernetics*, Vol. 28, No. 2 (2012), 129–140 (In Vietnamese).
- [6] Li R. P., Huang D., Reducts in incomplete decision tables, *Proceedings of the First international conference on Advanced Data Mining and Applications*, ADMA05 (2005), 165–174.
- [7] Pawlak Z., Rough Sets - Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
- [8] Skowron A., Rauszer C., The Discernibility Matrices and Functions in Information Systems, Intelligent Decision Support, *Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, (1992), 331–362.
- [9] Zou H. S., Zhang C. S., Efficient Algorithm for Knowledge Reduction in Incomplete Information System, *Journal of Computational Information Systems* 8: 6 (2012), 2531–2538.
- [10] Wei W., Liang J. Y., Qian Y. H., Wang F., Dang C. Y., Comparative study of decision performance of decision tables induced by attribute reductions, *International Journal of General Systems*, Vol. 39, No. 8 (2010), 813–838.
- [11] Zhou X.Z., Huang B., Rough Set-based Attribute Reduction under Incomplete Information Systems, *Journal of Nanjing University of Science and Technology*, 27(2003), 630–635.
- [12] Qian Y. H., Liang J. Y., Li D. Y., Wang F., Ma N. N., Approximation reduction in inconsistent incomplete decision tables, *Journal Knowledge-Based Systems*, Vol. 23(5) (2010), 427–433.

Exploiting Linked Open Data for Attribute Selection on Recommendation Systems

Xuan Hau PHAM ^{a,b} and Jason J. JUNG ^{b,c,1} and Hideaki TAKEDA ^c

^a*Department of Mathematics and Informatics, Quang Binh University
Dong Hoi, Vietnam*

^b*Department of Computer Engineering, Yeungnam University
Gyeongsan, Korea 712-749*

^c*National Institute of Informatics
2-1-2 Hitotsubashi, Chiyodaku, Tokyo, Japan*

Abstract. Recommendation systems have extracted items that users may be interested in. However, most of the recommendation applications have restricted on recommending only items in a specific domain (e.g., movies, books, and musics). In this paper, we propose a novel approach to enable the existing recommendation systems to extract addition items in various other domains. Thereby, this work is focusing on integrating all available Linked Open Data (LOD) for selecting more relevant attributes which can improve the performance of recommendation processes.

Keywords. Recommendation systems, Linked Open Data (LOD), Interlinking, Attribute selection, Attribute-Value

1. Introduction

Recommendation is a process to describe interactions between users and system. Users provide their feedback, and the systems recommend relevant items to user. Items are represented as a set of attributes and their values. We assume that the users select items, since they are interested in a certain set of attributes and values of these items [1,2,3,4].

Recommendation can extract these values to find out potential items which may be selected by user (i.e., not only movie but also other type of relevant items that have the same attribute values). Recommendation System (RecSys) shows not only relevant items that user may be interested in but also it provides full information about items to user. Almost RecSys often displays its specific item such as movie or book. Beside, the collecting and storing data is in a closed. It means that the system makes a private database and implements recommendation algorithms on this data. In fact, users may be interested in other types of items that they are the similar attribute values. For example, if user likes “Spider-Man” movie user may also be interested in many kinds of items that

¹Corresponding Author: Jason J. Jung, Department of Computer Engineering, Yeungnam University, Gyeongsan, Korea; E-mail: j2jung@gmail.com.

are related to it such as “Spider-Man” songs, “Spider-Man” books, “Spider-Man” clothes or “Spider-Man” toys. This is not only a barrier for user but is also a problem for the current systems.

Linked Open Data (LOD) describes the connecting data between different data sources by using the Resource Description Framework (RDF) and the Hypertext Transfer Protocol (HTTP) [5]. LOD is an effective mechanism to allow data in one data source to be linked to data in multiple data sources.

The goal of this work is to discover relevant information and related items from LOD based on attribute-value (AV) in order to diversify of information and types of item for RecSys. In our previous studies on AV-based RecSys, we have applied not only the rating from users but also additional information about items (e.g., movies) by extracting the attributes and values (e.g., IMDB) [2,3]. However, as much semantic information from Linked Open Data Cloud (LODC) is available, we have been considering how to exploit more information from LOD. In this paper, we present the extracting and integrating LOD with set of attribute values into RecSys. It is possible to construct open RecSys. This approach brings some benefits as follows:

- Developing recommendation mechanism,
- Diversifying different types of items,
- Providing more relevant information,
- Reducing stored data, and
- Reusing available data from different sources.

The outline of paper is organized as follows. In Section 2, we introduce some data sources for RecSys. In Section 3, we propose our approach to exploit LOD for attribute selection. In Section 4, the experimental results are presented and discussed. In Section 5, we present related work. Finally, in Section 6, we draw the conclusions of this work and describe our plan of future work.

2. Data Sources for Recommendation Systems

In this paper, we consider some data sources on the web as follows:

- IMDB² is a huge movie repository on the Internet. It contains movie information (e.g., URI, title, genre, actor, director, runtime,...). We have extracted movie data from IMDB and made our database. Assume that we have a list of URIs where identify link of movies.
- LinkedMDB³ describes movie information based on extracted data from IMDB and contains several links to movie-related data sources. In LinkedMDB, it contains entities, namely interlink. This is an important point that we will apply to matching. This will reduce the fuzzy string matching problems.
- DBpedia⁴ contains a lot of “things” data and several relevant information. It describes and represents related information by extracting them from Wikipedia.

²<http://www.imdb.com/>

³<http://www.linkedmdb.org/>

⁴<http://dbpedia.org/>

Table 1. Short description “Titanic” movie with LOD

Datasets	Property	Value
DBpedia	foaf:name	Titanic
	owl:sameAs	de.dbpedia.org/resource/Titanic_(1997)
	rdfs:comment	a list of comment
	dbpedia-owl:runtime	11640.00000 (xsd:double)
LinkedMDB	dc:date	1997-12-19
	rdfs:label	Titanic
	foaf:page	www.imdb.com/title/tt0120338/
Musicbrainz	dc:title	My heart will go on
	mo:composer	James Horner
	mo:lyricist	Will Jennings
	mo:recording	Celine Dion

- Musicbrainz⁵ is an open music encyclopedia that collects, and makes available to the public, music metadata. It captures information about artists, their products, and the relationships between them such as album title, track titles, the length of each track and so on.
- RottenTomatoes⁶ contains reviews, information of movies. It collects online reviews from users who belong to several writing guilds or film critic associations.

For example, Table 1 shows a short description about “Titanic” movie on LODs, e.g., DBpedia, LinkedMDB and Musicbrainz.

In most of the traditional RecSys (which are closed), the data repository is private. The system can only discover on this data to implement recommendation mechanism and algorithms. On the contrary, open RecSys can flexible connect data from different sources. It does not only extend and provide more information but also solves some problem such as “cold-start” and “data acquisition”.

3. Data Integration Based on LOD

The different types of items integration based on LOD can be considered via the connection by interlinks and same attribute values. Each type of the items can be described by a set of attributes:

- Movie: title, genre, actor, director, sound track, producer, company.
- Book: title, publisher, author, genre, edition.
- Music: track, singer, genre, composer, lyric, album.

The data integrations often rely on string matching. The data discovery in LOD allows systems to extract and retrieve semantic mappings by following RDF and interlinks. Thus, we have to attend to quality of matching. There are some issues related matching the movie title that are mentioned in [6]. It is considered as fuzzy strings matching problem. In our goal, we will exploit movie-related data from different data sources. We consider the following example: when we combine between IMDB and DBpedia based on movie attributes and movie attribute values then the result is not unique. By applying the following SPARQL query (i.e., matching movie title “Titanic”), we can obtain two movies (e.g., Titanic_(1953_film); Titanic_II; Titanic_(1997_film)).

⁵<http://musicbrainz.org/>

⁶<http://www.rottentomatoes.com/>

```

SELECT ?film ?film_label
WHERE
{
    ?film rdf:type dbpedia:Film .
    ?film rdfs:label ?film_label.
    FILTER (regex(?film, 'Titanic', 'i'))
}

```

We can find the movie title “The Russians Are Coming! The Russians Are Coming!” in LinkedMDB. This title is extracted by IMDB. The result is empty. However, when we find with the title “The Russians Are Coming, the Russians Are Coming” then the result is right. It means that if we apply the string matching based on movie titles then we get some limitations.

In order to solve above mentioned issues, we are based on the movie ID (MID) from IMDB to get interlink that contains URI of this movie on DBpedia. As we known, each movie on IMDB has a unique URI. The URI will identify a movie ID (e.g., ID of movie “Apollo 13” is “tt0112384”). We can use this ID to discover related information in order to avoid fuzzy strings matching.

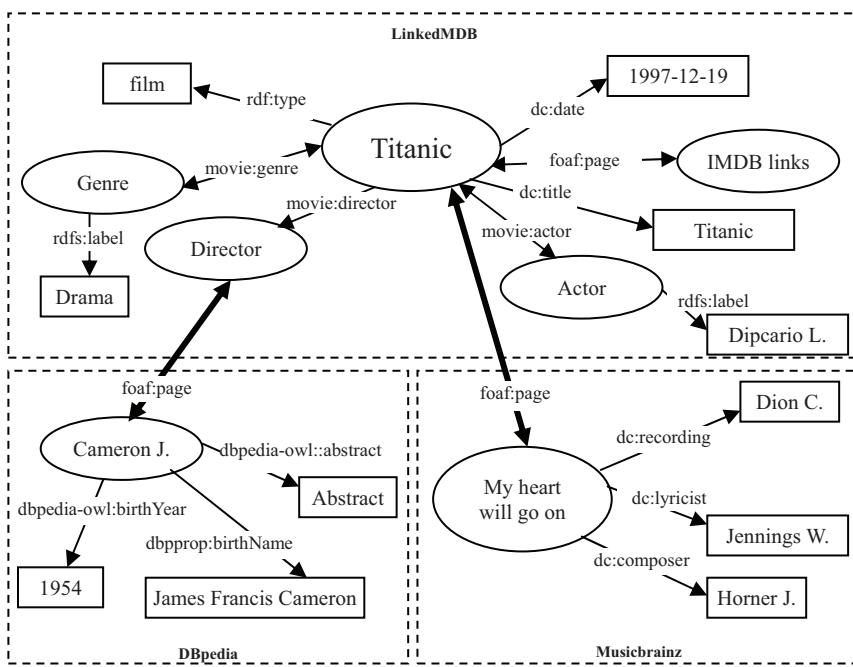


Figure 1. LOD Integration

In LinkedMDB, the data sources connections are identified by interlinks that have properties such as *foaf : page*, *owl : sameAs*. It has been discussed in [7]. The link contains *foaf : page* property that means it links to an external page with the same topic. A triple with an *owl : sameAs* property gets the information that two identifiers refer to the

same individual or entity. We will recognize URIs with these properties to make connection among datasets as Figure 1. We are based on the *foaf : page* property to get RDF that contain URI matching to IMDB. Also, matching two URI between LinkedMDB and DBpedia based on the *owl : sameAs* property.

We can see that each movie has a RDF which contains an IMDB's URI in LinkedMDB. This ensures a unique result. Therefore, instead of finding movie according to title, we will find movie according to movie ID. Besides, we also get the RDF which describes this movie on DBpedia.

Assume that in our database, we have a list of URIs. For example, we consider “Apollo 13” movie. It has URI : www.imdb.com/title/tt0112384/. The first one has to find URI of this movie in LinkedMDB

```
SELECT ?title ?link ?free WHERE {
    ?film rdf:type movie:film ;
    foaf:page ?link ;
    dc:title ?title ;
    dc:date ?d;
    owl:sameAs ?free .
FILTER (regex(STR(?free), 'dbpedia.org')
&& regex(STR(?link), 'tt0112384'))}
```

It contains URI of this movie on DBpedia. Second one will use this URI for extracting data from DBpedia.

```
SELECT ?name
WHERE { <http://dbpedia.org/resource/Apollo\_13\_\(film\)>
    dbpprop:music
    ?name }
```

From movie information, we can extract music information from Musicbrainz based on *Sound track* attribute of movie. For example, we can find “James Horner” composer information as follows:

```
SELECT ?artist ?name WHERE {
    ?artist a mo:MusicArtist
    .?artist foaf:name 'James Horner'
    .?artist foaf:name ?name
    .FILTER(regex(str(?name), 'James Horner', 'i'))}
```

4. Experimental Results

We have implemented with databases from IMDB, LinkedMDB and DBpedia: IMDB contains a huge amount of movies about 2 millions; LinkedMDB contains about over 40,000 movies; DBpedia also contains a lot of movies about 72,000. LinkedMDB provides the number of interlinks with *owl : sameAs* property link to DBpedia are about over 30,000 and *foaf : page* property link to IMDB and other data sources are about a half million. MusicBrainz provides information about 700,000 artists and over 11 million recordings.

Table 2. Statistics on Datasets

LOD	#interlink	
LinkedMDB	1692	84.6%
DBpedia	1619	95.7%
Musicbrainz	1569	78.5%

In order to illustrate our proposal in this paper, we have tested with small dataset about 2000 URIs from IMDB and we get the result as Table 2.

However, the matching interlinks have some false matches. In LinkedMDB, The movie “Titanic” contains interlink of IMDB but do not contain interlink of DBpedia. In some case, URI may be different representation. These movies cannot match URI of DBpedia. For example, the movie “Sophie Scholl: The Final Days” contains URI of DBpedia http://dbpedia.org/resource/Sophie_Scholl_a_The_Final_Days. This URI is not correct. The corrected URI is http://dbpedia.org/page/Sophie_Scholl_The_Final_Days

5. Related Work

In [8], they have proposed interlinking algorithms to integrate music data from Jamendo, Geonames and Musicbrainz datasets based on matching resources. Authors consider not only the similarity of the resources but also the similarity of their neighbors. They also show the analyzing user music collection as a dataset.

In [6], they have considered the matching of movie titles among several databases. They discussed about the identification such as different representation of the same movie, series movies, different spelling or writing in English of the same movie. They also provide the approximate string matching techniques to discover interlinking and the quality of the links is measured by different similarity functions.

In [9], they have applied LOD for collaborative filtering to build a Music RecSys. The system integrates different resources to a common vocabulary in order to make user-item matrix based on RDF graph of user-item relations. The system can extract data about users and items from LOD to fill up the matrix. They have integrated user profile and item profile from different data source to solve the new-user, new-item and sparsity problems of collaborative filtering.

6. Concluding Remarks

Reusing available datasets on the web is very essential for RecSys. In this paper, we consider exploiting LOD for attribute selection on RecSys. For this approach the system not only recommends a specific type of item but also other types of item. It is easy to help user in order to extend user requirements. However in this paper, we just focus on integrating LOD to provide more information and diversify types of items for RecSys. We have also used some datasets from different sources to demonstrate our approach.

In the future, we will extend this idea and reduce limitations by using more interlinks to improve quality of integration. We also consider the recommendation mechanism based on LOD.

Acknowledgement

This work was supported by Yeungnam University Research Grant in 2012.

References

- [1] Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6), 734–749 (2005)
- [2] Jung, J.J.: Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB. *Expert Systems with Applications* 39(4), 4049–4054 (2012)
- [3] Pham, X.H., Jung, J.J.: Preference-based user rating correction process for interactive recommendation systems. *Multimedia Tools and Applications* (to appear) DOI:10.1007/s11042-012-1119-8
- [4] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): *Recommender Systems Handbook*. Springer (2011)
- [5] Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *International Journal on Semantic Web and Information Systems* 5(3), 1–22 (2009)
- [6] Hassanzadeh, O., Consens, M.: Linked movie data base. In: *Workshop on Linked Data on the Web (LDOW 2009)* (2009)
- [7] Ding, L., Shinavier, J., Finin, T., McGuinness, D.L.: owl:sameas and linked data: An empirical study. In: *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line* (2010)
- [8] Raimond, Y., Sutton, C., Sandler, M.: Automatic interlinking of music datasets on the semantic web. In: *Linked Data on the Web (LDOW2008)* (2008)
- [9] Heitmann, B., Hayes, C.: Using linked data to build open, collaborative recommender systems. In: *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence* (2010)

Semantic Service Matchmaking for Ad Hoc Supply Chain Formation: a Network Analysis Approach

Duc Nguyen TRUNG ^{a,b} and Jason J. JUNG ^{b,1}

^a*Department of Information Technology, Vietnam Maritime University
 Hai Phong, Vietnam*

^b*Department of Computer Engineering, Yeungnam University
 Gyeongsan, Korea 712-749*

Abstract. Since the number of enterprises is getting larger, business collaborations become more complicated. Particularly, in a dynamic environment (e.g., supply chains), it has been difficult for the traditional enterprises to find and select relevant partners by using their *rigid* business relationship. In this paper, we propose a semantic service framework to conduct a supply chain formation, and more importantly, given a certain event, to choose the best *business paths* between two arbitrary enterprises. The participant enterprises are required to provide their own ontologies so as to obtain semantic matches between the services and justify their semantic interoperability. As a result, in terms of two indicators (i.e., precision and agility), we have shown that the proposed framework outperforms traditional enterprise collaboration schemes.

Keywords. Ad hoc supply chain, Enterprise alliance formation, Enterprise network, Service-oriented architecture (SOA)

1. Introduction

Flexible interoperability is an important hurdle to business collaboration. One of the main processes of the SCM system is to conduct efficient planning (e.g., scheduling and forecasting) methods to the businesses by gathering all available information and resources from other business partners. Thereby, these SCM systems have realized that the communication facilities between businesses should be efficiently supported to share information and resources. Then, currently, the resources on other businesses become somehow accessible and feasible by using several methods (e.g., ODBC, XML-RPC, and so on), as business information systems on supply chains have been interconnected with each other,

However, the problem is that only the standardized XML schema can allow to do this. It causes that the communication among business partners on a supply chain are limited (i.e., low flexibility). In *open* marketplace, particularly, the businesses are required

¹Corresponding Author: Jason J. Jung, Department of Computer Engineering, Yeungnam University Gyeongsan, Korea E-mail: j2jung@gmail.com

to consider more flexible interactions with businesses which are even unknown. More seriously, since the number of businesses in a marketplace is getting increased, some supply chains should be dynamically constructed for better performance.

Thus, in this paper, we focus on *enterprise alliance formation* in the dynamic environment. Enterprise alliance in online business is to integrate a set of virtual organizations (VO) which are closely linked with each other for achieving a certain unified business goal [1,2]. In order for the enterprise alliance to have better business strategies and tactics, it comes together to efficiently share not only services themselves but also various enterprise resources (e.g., experiences, knowledge, and useful competencies) whose cooperation is supported through computer networks. By taking into account the sequential links between services, enterprise alliance is regarded as *Service Chain Management* (SvCM) which enables service organizations to meet customer requests and to minimize costs through intelligent and optimized forecasting, planning and scheduling of the service chain, and its associated resources such as human, networks and other assets. Practically, SvCM can be applied to broad areas, covering field force and workforce automation, network and asset planning and also aspects of human resources systems [3], enterprise resource planning [4] and customer relationship management.

Moreover, in the dynamic environment, agility on discovering service chains is a crucial factor for enterprise alliance formation. Many events can be unpredictably occurred in many dynamic environment. Given a certain event, the SvCM system should be enough agile to build the most relevant service chains.

Thus, the main research questions of this study are *i*) how to discover meaningful relationships between services and *ii*) how to apply them to build the optimized service chain for a given event. Especially, in the context of value network [5], we have to consider more general case where a number of different businesses are participating in an enterprise alliance. Since such relationships between services will be exponentially increased, it is very difficult for human experts and administrators to manage and understand the services for a variety of service-oriented processes (e.g., building new services). It means that a service from a business has to be automatically compared with other services from different business for finding out how they are interrelated with each other (e.g., semantic relationships). Consequently, once we somehow have a comparison result attached with a certain relationship, a new service can be generated by composing two (or more) of the compared services.

The outline of this paper is as follows. In the following Section 2, we introduce a definition of service network. Section 3 presents network analysis methods for discovering useful structural patterns from service networks. In Section 4, we describe semantic interoperability dealing with the problem of semantic heterogeneity between businesses for service composition, and show a simple example. Section 5 will give an experimental results, and discusses some significant issues and compares our contributions with the previous studies, respectively. Finally, Section 6 draws our conclusions of this work.

2. From Enterprise Network to Service Network

Generally, enterprise networks for collaborations among online businesses tend to be usually static and consistently fixed. Such networks are made of not only standard middleware communication channels (e.g., EDI), integrated security packages (e.g., public

key infrastructure), but also database integration tools (e.g., IBM WebSphere Message Broker, SAP Exchange Infrastructure, Microsoft BizTalk Server and Oracle Enterprise Service Bus). These collaborations have been done with mutual agreements, market brokers and strategic partnerships.

Definition 1 (Enterprise agreement) *Given a set of enterprises B , various Service-Level Agreements (SLA) can be established between two arbitrary enterprises. According to levels, there are Corporate-level SLA (C_{Cor}), Customer-level SLA (C_{Cus}), Service-level SLA (C_{Ser}), and Multilevel SLA. Thus, in this paper, enterprise agreements can be represented as*

$$R_M = \{C_{Cor}, C_{Cus}, C_{Ser}\}. \quad (1)$$

In practice, enterprises can exploit Web Services Agreement Specification (WS-Agreement²) to represent them.

Definition 2 (Enterprise network) *An enterprise network of a enterprise alliance $\mathcal{N}_{\mathcal{B}}$ is represented as*

$$\mathcal{N}_{\mathcal{B}} = \langle \mathcal{B}, M, R_M \rangle \quad (2)$$

where \mathcal{B} is a set of enterprises which join to this enterprise alliance, and $M \subseteq |\mathcal{B}| \times |\mathcal{B}|$ means a set of direct partnerships between enterprises which are manually established in real world. Additionally, R_M is a set of agreement types made between the corresponding enterprises.

In this paper, we assume that an enterprise alliance is based on SOA computing environment. It means that these enterprises in the enterprise alliance have to describe the services that they open and provide to any other enterprises. These service descriptions are advertised to the others. There should be a standard language (e.g., WSDL) to make others understand.

Furthermore, this services can contain semantic information (e.g., ontological elements) extracted from their local ontologies. This process is also referred to semantic annotation of business processes [6,7]. Even though there are many different definitions on ontologies, in this study, we choose a simplistic approach on ontology engineering by merging a set of faceted taxonomy [2,8]. The faceted taxonomy can include various domain knowledge (e.g., product catalogue) which is composed of a set of classes.

Definition 3 (Faceted taxonomy [2]) *Given an enterprise $B_k \in \mathcal{B}$ participating in an enterprise alliance $\mathcal{N}_{\mathcal{B}}$, a faceted taxonomy FT_k of B_k can be defined as a set of subclass assertions between classes \mathcal{C}_k . Hence, FT_k is given by*

$$FT_k = \{\langle c_i, subc, c_j \rangle | c_i, c_j \in \mathcal{C}_k, c_j \sqsubseteq c_i\} \quad (3)$$

where c_i means a superclass of c_j .

²<http://www.ogf.org/>

Once the faceted taxonomies of the parties in the enterprise alliance are collected, they are merged with each other and regarded as an enterprise alliance ontology.

Definition 4 (Ontology) An ontology $\mathcal{O}_{\mathcal{B}}$ of an enterprise alliance $\mathcal{N}_{\mathcal{B}}$ is built by aggregating a set of faceted taxonomies. Suppose that a set of enterprises $\mathcal{B} = \{B_1, \dots, B_N\}$ be comprised in an enterprise alliance. Thus, the ontology $\mathcal{O}_{\mathcal{B}}$ is simply formulated by

$$\mathcal{O}_{\mathcal{B}} = \bigcup_{B_k \in \mathcal{B}} FT_k. \quad (4)$$

Surely, since we want to remove the duplications, there should be some process to discover alignments between the faceted taxonomies. More importantly, given two taxonomies (i.e., FT_a and FT_b) from two arbitrary enterprises (i.e., B_a and B_b), domain experts can manually assert alignments

$$\mathcal{A}_{\mathcal{B}} = \{\langle c_p, rel^*, c_q \rangle | c_p \in FT_a, c_q \in FT_b\} \quad (5)$$

where rel^* indicates the semantic relationship (e.g., equivalence and subsumption) declared by the human expert. The mapping can be expressed with various relations between classes in different faceted taxonomies.

Thus, by using the ontology, the services provided from the enterprise alliance can be semantically annotated.

Definition 5 (Service) A service can be simply described by semantic annotation process. Thus, we assume that a service s from B_i is represented as

$$s = \{c_k | c_k \in \mathcal{O}_{\mathcal{B}}\} \quad (6)$$

where these concepts are derived from the enterprise ontology $\mathcal{O}_{\mathcal{B}}$.

Practically, there have been many kinds of software APIs to annotate Web services. In this work, we have employed SAWSDL4J API provided from METEOR-S framework³. Consequently, services with semantics are expected to show better interoperability on SOA environment. More detail on this issue will be explained in Section 4.

Given a certain event (or goal), decision makers have to figure out which services are necessary to execute, and how the services are sequentially connected. Thus, we want to build a service chain to represent all possible relationships between services provided by the enterprise alliance.

Definition 6 (Service chain) Given an enterprise alliance, a service chain $\mathcal{N}_{\mathcal{S}}$ is defined

$$\mathcal{N}_{\mathcal{S}} = \langle S, E, R \rangle \quad (7)$$

where S is a set of services supplied by the enterprise alliance, and $E \subseteq S \times S$ is a chain matrix, meaning a set of relationships between services. Additionally, $R = \{\equiv, \sqsupseteq, \sqsubseteq, \perp, \top\}$ is a set of semantics for describing the semantics of service relationships.

³<http://lsdis.cs.uga.edu/projects/meteor-s/>

By referring to the semantic annotations of the services, we can determine whether two services are semantically related with each other or not. Practically, it is difficult for the software systems to automatically discover the relationships between services. As alignments between faceted taxonomies can be done by human experts, the service relationships can be attached by human experts by referring to semantics from ontologies as well as their own experiences and knowledge.

3. Service Chain Analysis

Various measures have been proposed on social networks between people designed from social network analysis [9] and from semantic social network [10]. This is based on topological analysis on the graph-structured information spaces to discover hidden knowledge underlying the networks. Eventually, we can realize who is the most important person in the social network.

In this context, once we have built a service chain, a number of those social network analysis methods can be exploited. Then, we can be aware of importance of individual services on the service chain. This paper claims that this information is useful for formating enterprise alliances. Note that these measures apply only if the service chain is connected with direction. These measures are often normalized (between 0 and 1) but we present their simplest form.

Shortest path distance (SPD) Given two arbitrary services s and s' in a service chain, we can find out a shortest path SP between them, and also measure the geodesic distance of the path. It is denoted as $SPD(s, s')$. It can be computed by repeating multiplying the chain matrix E . The larger SPD value between two services is indicating the poorer relatedness between them.

Closeness centrality The inverse of average length of the shortest path between a service s and any other services in the service chain is given by

$$Closeness(s) = \frac{1}{\sum_{(s',e,r) \in \mathcal{N}} SPD(s, s')} \quad (8)$$

where \mathcal{N} indicates a service chain of the given enterprise alliance.

Betweenness centrality [11] The proportion of shortest paths between two services which contains a particular service (this measures the power of this service) is given by

$$Betweenness(s) = \sum_{s' \neq s \neq s'' \in \mathcal{N}} \frac{\sigma_{s',s''}(s)}{SP(s', s'')} \quad (9)$$

where $\sigma_{s',s''}(s)$ (by Bellman criterion [12]) indicate the number of shortest paths $p(s', s'')$ in $SP(s', s'')$ that service $s \in \mathcal{N}$ lies on.

Hub and authority There are different but interrelated patterns of power: *i*) authorities that are referred to by many good hubs, and *ii*) hubs that refers to many good authorities. The highest authorities are those which are referred to by the highest hubs and the highest hubs that those which refers to the highest authorities. Kleinberg

[13] proposes an iterative algorithm to measure authority and hub degree of each entity in interlinked environment. Hence, given initial authority and hub degrees of 1, the degrees are iteratively computed by

$$Hub_{t+1}(s) = \sum_{\langle s', e, r \rangle \in \mathcal{N}^i} Auth_t(s') \text{ and} \quad (10)$$

$$Auth_{t+1}(s) = \sum_{\langle s', e, r \rangle \in \mathcal{N}^i} Hub_t(s') \quad (11)$$

where $SPD(s, s') = 1$.

Similarly to betweenness, the hub weight indicates the structural position of the corresponding service. It is a measure of the influence that services have over the spread of information through the service chain.

4. Interoperability by Enterprise Alliance

Semantic heterogeneity problem between businesses is caused by several reasons. Formation of the knowledge are semantically distinct with each other, because the knowledge are designed by experiences and heuristics of the local experts (or administrators). It means that semantic information extracted from the knowledge may be heterogeneous with the others. Such heterogeneities are caused by the difference of not only the terminologies (e.g., synonyms and antonym), but also, more importantly, the knowledge structures (e.g., database schema [14] and ontologies [2]). Consequently, it is difficult for the enterprises to be integrated, and more importantly, it is impossible for the enterprise alliances to automatically achieve *strategic* cooperations (e.g., *i*) business rules, e.g., strategies and policies, and *ii*) hierarchical taxonomies for describing the resources) with heterogeneous enterprise alliances.

In order to overcome this drawback, we have focused on semantic interoperability between virtual enterprises [2]. A large number of enterprises have been inter-related with the others in a same enterprise alliance or different enterprise alliances for performing ad-hoc (or real-time) collaboration. In order to provide efficient interoperability between the enterprises, the heterogeneities between the corresponding ontological knowledge structures have to be dealt with. Thereby, we have to consider efficient alignment method to resolve their conflicts. While intra-alignment is a process merging all local ontologies into an organizational ontology, inter-alignment is a process mapping all semantic correspondences between two organizational ontologies.

We have proposed an efficient method to build an integrated enterprise alliance by mapping heterogeneous ontologies of enterprises, i.e., maximizing the summation of partial similarities between a set of possible pairs of classes. The partial similarity can be calculated by comparing both set of instances in the classes. After both ontologies are aligned at conceptual level, and the source ontology instances are transformed into the target ontology entities according to those semantic relations.

4.1. Discovering semantic relationships between services

For unveiling the relationships between services, we have to figure out the relationships between the corresponding descriptions (i.e., concepts). Thus, we have to conduct ontology matching process. After ontology matching process⁴, the alignments between heterogeneous ontologies can be represented as a set of pairs of concepts from two different ontologies. We refer these concept pairs to correspondences (e.g., equivalence or subsumption).

Definition 7 (Alignment) *Given two ontologies FT_i and FT_j , the alignments between two ontologies are represented as a set of correspondences $CRSP_{ij} = \{\langle c, rel, c' \rangle | c \in FT_i, c' \in FT_j\}$ where rel means the relationship between c and c' , by maximizing the summation of class similarities.*

Finally, alignment process makes heterogeneous enterprise alliances interoperable (even partially) among them. For example, local users in an enterprise alliance can easily and transparently access to the other enterprise alliances. To do so, enterprise alliances have to conduct the ontology matching process in advance. Suppose that a set of enterprise alliances $\mathcal{B} = \{B_1, \dots, B_N\}$ should be interoperable with each other. Alignment process can find out the correspondences between all pairs of ontologies, i.e., B_i obtains $N - 1$ sets of correspondences.

Table 1. Service relationship discovery by semantic matching process

Scope	Service description	Semantic relationships
In a same enterprise alliance	$d_s = d_{s'}$	$s \equiv s'$
	$d_s \subseteq d_{s'}$	$s \sqsubseteq s'$
	$d_s \cap d_{s'} \neq \emptyset$	not decidable
In a different enterprise alliance	$d_s = d_{s'}, d_s \subseteq d_{s'}, d_s \cap d_{s'} \neq \emptyset$	not decidable
	$\{\langle c, \equiv, c' \rangle c \in d_s, c' \in d_{s'}\} \sqsubseteq CRSP_{ij}$	$s \equiv s'$
	$\{\langle c, \sqsubseteq, c' \rangle c \in d_s, c' \in d_{s'}\} \sqsubseteq CRSP_{ij}$	$s \sqsubseteq s'$

Most importantly, given two services s from B_i and s' from B_j in a semantic SvCM, the relationship between both of them should be discovered. Table 1 shows a simple example of patterns for establishing relationships between services. Certainly this table can be expanded, according to the strategies on the SvCM.

4.2. Example

In this section, we want to show a simple example based on service network analysis methods. While on a conventional marketplace with online and offline enterprises, the enterprises are interlinked with each other by mutual agreements and contracts (e.g., supply chains), we have been considering integrating and merging the link-based structures from several business sectors. When we need to find out the best service chain for achieving a certain goal (i.e., sequentially aggregating enterprises until customers), the best one

⁴We skip ontology matching processes. Please refer to other literatures [17] for more details.

should be selected out of a set of all possible service chains by taking into account the semantic interoperability between the enterprises.

Moreover, if they have semantic-based information systems on open networks, we can obtain semantic relationships between such enterprises located in different sectors. For example, by matching pairs of ontologies,

- $S(O(\text{Medical Equipment Supplier}), O(\text{Health Wholesalers})) = 0.64$
- $S(O(\text{Other Equipment Supplier}), O(\text{Pharmaceuticals Supplier})) = 0.33$

we can realize that among all possible service chains from “R&D Laboratories” to “Customers,” “Medical Equipment Supplier” and “Health Wholesalers” is more closely related with each other, compared to “Other Equipment Supplier” and “Pharmaceuticals Supplier.”

5. Discussion and Related Work

Here, we want to put some discussion about the Web Services. Web Services have been regarded as one possible way of realizing the technical aspects of the so-called SOA (service-oriented architecture). These services can be new applications or just wrapped around existing legacy systems to make them SOA-enabled. Common technologies for developing web services are WSRF⁵, SOAP⁶, UDDI⁷ and WSDL⁸. Furthermore, when using these technologies XML is a basic technology for developing web services this way. For reasoning aspects, a Web Service is interesting if several reasoning components are available and accessible through the use of indexes possibly managed by other entity (a broker). A user is able to request a specific reasoning component by checking the indexes of the storage. For this three instances can be identified, a service consumer, a service provider and a Service Broker (storage of indices).

There have been several important research issues on dealing with semantic matching between information systems. For doing this, many studies have been proposed to provide interoperability by discovering and integrating local knowledge structures between VOs [15]. They can be briefly noted into three issues;

- Incremental discovery of local knowledge [16],
- Knowledge matching (including schema and ontology matching) [17], and
- Interoperability via third-party platforms, e.g., service-oriented architecture (SOA) [18].

Moreover, human understandability is also important problem for taking care of a large-scaled services and resources. In fact, in this work, we are focusing on supporting local users (e.g., decision makers) through aligning the ontologies applied to annotate (or classify) the services on enterprise alliances. It means the local users in a certain enterprise alliance can access to the other enterprise alliances which are not familiar with them. Unlike a centralized portal systems (e.g., meta search engines), the local users can be provided with a set of concept mapping extracted from direct alignments,

⁵Web Service Resource Framework

⁶Simple Object Access Protocol

⁷Universal Description, Discovery and Integration

⁸Web Service Description Language

so that they can deploy meaningful translation services (e.g., query expansion [19] and transformation).

We can think of some related work which should be compared with the proposed work. Third party logistics [20,21] is an important domain to consider the service integration and composition for optimal solutions. Similarly, on-demand e-supply chain integration [22] has proposed a real-time approach to solve the service heterogeneity problems.

These systems have been employed to a various domains like geographical location-based system [23], competitive partner selection [3], and collaborative service chain [4].

6. Conclusions and Future Work

Services have been regarded as a key factor on business success. In the context of information engineering domain, a large amount of information from (and to) enterprises should be efficiently processed and manipulated to maximize the values by integrating relevant businesses together. Particularly, service-oriented architecture (SOA) is regarded as an efficient platform to exchange services (e.g., publishing and subscribing services) between enterprises. Within SOA platforms, XML-based standards have been employed by the enterprises.

Recently, service-dominant logic [24] has been significantly emphasized on many researchers in various domains of management, social, and engineering science [25]. This paper is a theoretical paper for introducing a basic idea of service network analysis. We have presented a conceptual framework to integrate multiple service networks which had been isolated only in individual business sectors into a global service network. Hopefully, the services can be annotated with business ontologies, so that the ontology alignment algorithms are efficiently applied to find out the relationships between services. In addition, we want to note that the services mentioned in this paper is derived from online enterprises as well as from offline enterprises. More importantly, traditional social network methods can be applied to understand the topological patterns from the integrated service networks.

As future work, we want to describe research limitations and problems that we have been realizing during this study as follows;

- Legacy problem: It is difficult for offline legacy enterprises to put semantics into them. We are expecting some machine learning approached to deal with this issue.
- Semantic description of services: There have been several service ontologies and service metadata.
- Human understandability: A study and system on service visualization or service network visualization are needed to increase understandability of human users.

Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012-013-1582).

References

- [1] Vicky Manthou, Maro VLachopoulou, D.F.: Virtual e-chain (vec) model for supply chain collaboration. *International Journal of Production Economics* 87(3), 241–250 (2004)
- [2] Jung, J.J.: Taxonomy alignment for interoperability between heterogeneous virtual organizations. *Expert Systems with Applications* 34(4), 2721–2731 (2008)
- [3] Sarkis, J., Talluri, S., Gunasekaran, A.: A strategic model for agile virtual enterprise partner selection. *International Journal of Operations & Production Management* 27(11), 1213–1234 (2007)
- [4] Stubbings, P., Virginas, B., Owusu, G., Voudouris, C.: Modular neural networks for recursive collaborative forecasting in the service chain. *Knowledge-Based Systems* 21, 450–457 (2008)
- [5] Basole, R.C., Rouse, W.B.: Complexity of service value networks: Conceptualization and empirical investigation. *IBM Systems Journal* 47(1), 53–70 (2008)
- [6] Verma, K., Sheth, A.P.: Semantically annotating a web service. *IEEE Internet Computing* 11(2), 83–85 (2007)
- [7] Jung, J.J.: Semantic business process integration based on ontology alignment. *Expert Systems with Applications* 36(8), 11013–11020 (2009)
- [8] Tzitzikas, Y., Analyti, A., Spyros, N., Constantopoulos, P.: An algebra for specifying valid compound terms in faceted taxonomies. *Data & Knowledge Engineering* 62(1), 1–40 (2007)
- [9] Wasserman, S., Faust, K.: *Social Network Analysis*. Cambridge University Press (1994)
- [10] Jung, J.J.: Service chain-based business alliance formation in service-oriented architecture. *Expert Systems with Applications* 38(3), 2206–2211 (2011)
- [11] Freeman, L.: Centrality in social networks: Conceptual clarification. *Social Networks* 1, 215–239 (1979)
- [12] Brandes, U.: A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* 25(2), 163–177 (2001)
- [13] Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5), 604–632 (1999)
- [14] Hull, R.: Managing semantic heterogeneity in databases: a theoretical prospective. In: *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS 97)*. pp. 51–61. ACM Press, New York, NY, USA (1997)
- [15] Castano, S., Ferrara, A., Montanelli, S.: Matching ontologies in open networked systems: Techniques and applications. *Journal of Data Semantics* 5, 25–63 (2006)
- [16] Jung, J.J.: Ontological framework based on contextual mediation for collaborative information retrieval. *Information Retrieval* 10(1), 85–109 (2007)
- [17] Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. *Journal of Data Semantics* 4, 146–171 (2005)
- [18] Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. *IBM Systems Journal* 44(4), 887–904 (2005)
- [19] Qiu, Y., Frei, H.P.: Concept based query expansion. In: Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. Pittsburgh, PA, USA, June 27 - July 1. pp. 160–169. ACM (1993)
- [20] Ying, W., Dayong, S.: Multi-agent framework for third party logistics in e-commerce. *Expert Systems with Applications* 29(2), 431–436 (2005)
- [21] Yao, Y., Palmer, J., Dresner, M.: An interorganizational perspective on the use of electronically-enabled supply chains. *Decision Support Systems* 43(3), 884–896 (2007)
- [22] Wang, M., Liu, J., Wang, H., Cheung, W.K., Xie, X.: On-demand e-supply chain integration: A multi-agent constraint-based approach. *Expert Systems with Applications* 34(4), 2683–2692 (2008)
- [23] Yue, P., Di, L., Yang, W., Yu, G., Zhao, P.: Semantics-based automatic composition of geospatial web service chains. *Computers & Geosciences* 33(5), 649–665 (2007)
- [24] Lusch, R.F., Vargo, S.L., Wessels, G.: Toward a conceptual foundation for service science: Contributions from service-dominant logic. *IBM Systems Journal* 47(1), 5–14 (2008)
- [25] Larson, R.C.: Service science: At the intersection of management, social, and engineering sciences. *IBM Systems Journal* 47(1), 41–51 (2008)

A Combination of Business Rule and Modeling Languages for Semantic Business Processes Modeling

Chi P. T. TRAN^{a,1} and Hanh Huu HOANG^b

^a *Phu Xuan Private University, Vietnam*

^b *Hue University, Vietnam*

Abstract. With increasing requirements for business agility and cost pressures on information technology (IT), Business Process Management (BPM) is asked to move towards “Dynamic BPM” and “Intelligent Case Management” instead of freezing process flows in hard to change IT solutions. Business rules and business processes are essential artifacts in defining the requirements of a software system especially in Business Process Management Systems. Although business rules and business process are considered important ingredients of dynamic BPM, both of which are not enough for representing all business constructs. Business Process Model and Notation (BPMN) developed by OMG group is one of the newest standards among business process modelling languages. However, there are some critical aspects of business that BPMN should be extensive integration with business vocabularies and constraints. Semantics of Business Vocabulary and Business Rules (SBVR), another OMG standard, provides a meta-model for the semantic and declarative models of business vocabulary and business rules. In this paper, we have a comparison among the business rule and business process languages in order to show that no single language is internally complete with respect to the Bunge-Wand-Weber representation model. We also show BPMN and SBVR are better suited for combining process and rule modelling than any of these modelling languages used independently.

Keywords. BPM, Business rules, Business rule languages.

1. Introduction

Business Process Management’s efforts are to bring the business and IT communities. With increasing needs for business agility and cost pressures on IT, Business Process Management (BPM) is asked to move towards “Dynamic BPM” and “Intelligent Case Management” instead of freezing process flows in hard to change IT solutions. Business modelling takes an important role in BPM lifecycle in building conceptual models for various aspects of business, e.g., structure, processes, constraints, resources, etc. Business process and business rule modelling are the most popular approaches for modelling the behaviour of business. In process modelling, operational behaviour of enterprise is represented through process models while rule modelling defines rules as

¹Corresponding Author: Chi P. T. Tran, Phu Xuan Private University, 176 Tran Phu Street, Hue City, Vietnam; E-mail: phuongchi0910@gmail.com

requirements for business processes where they describe the constraints must hold for business processes that implement these business rules.

Business Process Model and Notation (BPMN) [1] developed by OMG group is one of the newest standards among business process modelling languages. It represents objects of the real world and seeks to eliminate the existing gap between modelled real world business processes and computerized processes. Moreover, it narrows the communication gap between business people and IT experts. According to the researches carried out by different scientists and practitioners, BPMN models the dynamic of business processes in a very proper way; however, there are some critical aspects of business that BPMN should be extensive integration with business vocabularies and constraints.

Semantics of Business Vocabulary and Business Rules (SBVR) [2], an OMG standard, provides a meta-model for the semantic and declarative models of business vocabulary and business rules. SBVR was welcomed by both business and IT sectors. The vision of SBVR is to express business knowledge in a controlled natural language, which is unambiguous and understandable to human as well as to computer systems. Logical formulation of SBVR facilitates IT people to interpret these models generated by business people. Benefits of SBVR are its declarative nature, rule modelling approach, natural language representation and formal logical backbone.

Although both the process and rule modelling approaches have different benefits in terms of semantic representation, little work has been done to understand the relationship between them. Representational analysis done by Michel et al. [3] shows that any single approach is not capable of representing all business constructs. Rule modelling approach focuses on decision points which regulate the business. On the other hand, the process modelling approach tries to minimize the amount of work required in processing but ignores decision points. In order to overcome this problem, many researchers have proposed the combination of a business rule and a business process modelling language covers maximum business representational constructs.

For the purpose of dealing with the problem of combination of two languages, this paper makes a review on business process and business rule modelling languages which leads to the analysis of disadvantages of these languages in modelling business process. Concretely, we conduct a comparison on these modelling languages according to BWW (Bunge-Wand-Weber) model which consists of over forty ontological constructs. The objective of this comparison is to indicate that business rules or business process approach or process based approach is not good enough for representing all business constructs. So as to cover the business construct, the integration of business rules and business processes has been proposed.

This paper is structured as follows: section 2 makes a comparative overview on business rule and business process modeling languages; section 3 gives the reason why integrating of business rules and business processes and a case study for this combination is shown in section 4; section 5 is the related work in combining of business process and business rule languages; section 6 concludes overall the paper.

2. A Comparative Overview on Business Rule and Process Modelling Languages

2.1. Business Rules

There are some different definitions of business rules that can be found. Business rules are atomic, formal expressions of business policies, business regulations and common-sense constraints [4]. A business rule is a statement that aims to influence or guide behaviour and information in an organization [3]. However, the definition most commonly used is the definition of the Business Rules Group which is a group of IT-professionals and one of the developers of the SBVR standard (Business Rules Group). According to these authors, a business rule is: “a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behaviour of the business”. From these definitions, we have found that all these definitions have focused on the following factors of a business rule: is a statement that defines or guide the aspect of the business. The definition of Goedertier et al. [4] adds to business rules have to be atomic, and formal expressions. Atomic means that the statement cannot be broken down or decomposed further into other business rules, without losing its meaning. Formal means that a business rule specification must be transformed automatically. In others word, business rule specification has to be parsed by machine, and the machine must understand what is in the specification in order to do something with it. Therefore, it must be defined in a formal language.

2.2. Business Rule Modelling Languages

As mentioned above, computers can understand the business rules specification if and only if they are defined by a formal language called business rule specification languages. A variety of rule languages have been developed over the past decade. In this section, we give a brief introduction about business rule modelling languages.

Object Constraint Language (OCL) is a business modelling language initially developed within IBM insurance division. It is a language for describing rules that apply to Unified Modelling Language (UML) models. A class diagram in UML for example, may not be refined enough to describe object query expressions or other relevant constraints about objects. OCL provides that additional support and can actually be used with any Meta-Object Facility (MOF) Object Management Group [5] meta-model.

Semantic Web Rule Language (SWRL) is a submission to the W3C trying to combine the rules (RuleML) and ontologies (OWL-DL and Lite) in May 2004. Rules in SWRL are expressed in terms of OWL constructs such as individuals, properties, literals, and classes. Rules are written as antecedent-consequent pairs [6]. While SWRL is not standardized, it is a widely-used (or more modestly saying – widely-considered) language supported by a few commonly-used reasoners.

REWERSE Rule Markup Language (R2ML) is a general rule and rule modelling language developed by REWERSE I2 working group. Originally designed as an XML based rule interchange language, it was developed on a range of different rule modelling languages. R2ML integrates the Object Constraint Language (OCL), Semantic Web Rule Language (SWRL), and the Rule Markup Language (RuleML) (Reverse Working Group I1, 2006). As an interchange language (similar to RIF),

R2ML allows to preserve the structure of each of the language constructs without the need for translating the rule expressions to completely new language [7].

Rule Interchange Format (RIF) numerous rule languages exist to date, therefore there is a need to develop systems for exchanging rules between them. Rule Interchange Format (RIF) attempts to address this issue. Since 2005, the World Wide Web consortium [6], and its Rule Interchange Format Working Group, has been working on a development of a standard for exchanging rules between existing rule systems (W3C-RIF Working Group, 2010).

The Rule-based Business Process Modeling Notation (rBPMN) language is a product of integration of BPMN and R2ML, and it is defined by weaving the elements of the BPMN and R2ML abstract syntaxes (metamodels). rBPMN has been designed to support a rule-enhanced process-oriented modelling of service orchestrations and choreographies. More details about this language can be found in [8].

Semantics of Business Vocabulary and Business Rules (SBVR) [2] is an OMG standard for expressing business knowledge in the language that is primarily understandable to the business people. SBVR is aimed at helping business people understand models with no special knowledge in modelling notations or IT skills. The most common way of expressing business vocabularies and business rules is by using textual specifications rather than some diagrams. While diagrams are helpful when one needs to see how business concepts are related, they are not well suited for defining vocabularies and expressing rules. SBVR uses controlled natural language for business model specification. SBVR realizes the core principle of the Business Rules Approach at the business level, so called business rules “mantra” that has been introduced by Business Rules Group in 1995.

2.3 Business Process Modelling

Business process models take a pivotal role in the business process management lifecycle. Thereby, the great diversity of business process modelling languages has already led to researchers attempting to chart all these languages [9]. Particularly, a research [10] gives an excellent overview of these languages. We will provide a less elaborate overview and only describe those languages that have played an important role in business process modelling.

Flowcharts are the oldest and most basic process related modelling methodology known, with their first reported occurrence dating back to the early twenties, where they were used by mechanical engineers to describe machine behaviour. Besides the simple set of constructs, flowcharts offer little to none methodological support.

A Petri net is a formal modelling language for the description of concurrent processes. Unlike other methodologies, Petri nets have a strict mathematical definition of their execution semantics, which means that all well-formed Petri nets can be interpreted and executed by a machine. Moreover, the mathematical basis of Petri nets makes them suitable for various kinds of automated analysis. With respect to business process modelling, Petri nets have been used to translate non-formal process models into formal process models for the purpose of analysing or simulating these processes [11] [12].

A Role Activity Diagram (RAD) is an element of the STRIM business process modelling methodology developed by Praxis Plc for the elicitation, modelling and analysis of business processes. As the name suggests, RAD are very similar to UML activity diagrams, with the main differences being that RAD emphasizes

responsibilities, while UML activity diagrams emphasizes orchestration of the activities (e.g. aspects regarding the sequential or parallel execution of activities). There are no formal semantics underlying a RAD as these models are aimed at facilitating shared understanding among stakeholders, rather than providing a basis for process simulation or execution. However, some research has aimed to formalize RADs by translating them to other representations such as Petri Nets in order to perform simulations [13].

An Event-driven Process Chain (EPC) is a business process modelling methodology aimed at creating business understandable models. The emphasis on logical connectors and functions makes EPC seem technically oriented, while it is in fact aimed at the business stakeholders, rather than the IT stakeholders. There is only little formal semantics underlying EPC, although some research has been performed on formalizing EPC.

The Integrated Definition for Functional Modeling (IDEF) is a series of modelling languages originally designed to be used in the field of software engineering. IDEF family has grown to a set of sixteen modelling languages, each addition simply named IDEF0, IDEF1 and so on. However, only the first five IDEF languages have matured into well-accepted modelling languages, while the rest were never developed any further than their initial definition.

One of the most popular business process modelling languages is **BPMN (Business Process Modeling Notation)**. BPMN is a standard for business process modelling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique very similar to activity diagrams from Unified Modeling Language (UML). The primary goal of BPMN [1] is to provide a standard notation readily understandable by all business stakeholders. These include the business analysts who create and refine the processes, the technical developers responsible for implementing them, and the business managers who monitor and manage them.

Business process languages appear as Graph-based languages (e.g. BPMN, EPC), Net-based languages (e.g. Petri-nets, flow nets) and Workflow Programming Languages. Such languages, in general, are considered to be of the procedural modeling type, in that they focus on specifying the step-by-step activities that are required to take place in order to perform an action. While they do not provide the same level of precision or formalism as rule modeling languages, their strengths stem from their relative user friendliness and structural properties. Compared to procedural languages, most of the declarative languages cover only a specific part of a process model. Sadiq et al [14] proposed a constraint specification framework for flexible business processes. This approach can be viewed as a starting point for declarative modelling as only a part of the process model is described declaratively. Van der Aalst et al. [15] have given a declarative paradigm to model flexible business processes where execution of the process depends upon the current case. However, it is very restricted due to data-driven approach and does not cover all possible real scenarios.

In order to analyse the selected languages, we followed a reference methodology for conducting ontological analysis [16]. We examined the BWW (Bunge-Wand-Weber) model which consists of over forty ontological constructs (including constructs in sub categories) [3]. After we defined main ontology constructs in BWW model, we then started the process of identifying corresponding constructs in the modelling language. Based on [17], we divided BWW model into four main clusters: Thing, State, Event, and System. We then defined the subgroups of each cluster (see Table 1). We

then performed a representational analysis to compare each of the language constructs with constructs of a BWW model [17] and vice versa.

Table 1. Comparison on the business rule and business process modeling languages

BWW construct	OCL	SWRL	R2ML	RIF	rBPMN	SBVR	BPMN
Thing	+	+	+	+	+	+	+
Property	+	+	+	+	+	+	+
Class	+	+	+	+	+	+	+
Kind	-	-	-	+	+	-	+
State	+	-	+	+	+	-	-
Conceivable state space	-	+	-	-	-	-	-
Lawful state space	-	-	-	-	-	-	-
State law	+	-	+	+	+	+	-
Stable state	-	-	-	-	-	-	-
Unstable state	-	-	-	-	-	-	-
History	-	-	-	-	+	-	-
Event	-	-	+	-	+	-	+
Conceivable event space	-	-	+	-	+	-	-
Lawful event space	-	-	+	-	+	-	-
External event	-	-	-	-	+	-	+
Internal event	-	-	-	-	+	-	+
Well-defined event	-	-	-	-	+	-	+
Poorly-defined event	-	-	-	-	+	-	+
Transformation	+	+	+	+	+	-	+
Lawful transformation	+	+	+	+	+	-	+
Coupling	-	-	-	+	+	-	+
System	-	-	-	-	+	+	+
System environment	-	-	-	-	+	-	+
System composition	-	-	-	-	+	+	+
System decomposition	-	-	-	-	+	+	+
System structure	-	-	-	-	-	-	-
Subsystem	-	-	-	-	+	-	+
Level struct	-	-	-	-	+	-	+
Excess	7/28	6/28	10/28	9/28	23/28	7/28	18/28

The comparison indicates that business rule or business process approach or process-based approach is not good enough for representing all business constructs. In order to cover the business construct, the integration of business rules and business processes has been proposed.

3. Integration of Business Rules and Business Processes

3.1. Motivation

Business rules can be seen as requirements for business processes where they describe the constraints must hold for business processes that implement these business rules. Both the process modelling and rule modelling approaches have different benefits in terms of semantic representation. Representational analysis done by Michel et al [3], shows that any single approach is not capable of representing all business constructs because of the following reasons:

- Rule modelling approach focuses on decision points which regulate the business. Business rule modelling languages are typically based on formal logic and have strong precise expressive power. In general, they belong to declarative modelling category in which they focus on specifying a process as a set of process elements and declarative statements representing constraints over them.
- Process modelling approach is considered to be of the procedural modelling type in which they focus on specifying the step-by-step activities and defines a process as an exact sequence of process elements (tasks, events, etc.) and routing elements (gateways).

Table 2. Comparison of Procedural modelling and declarative modelling

	Procedural Modelling	Declarative Modelling
Business concerns	Implicit	Explicit
Rule enforcement	What, When and How	What
Communication	What, How	What
Execution scenario	Design-time	Run-time
Execution mechanism	State-driven	Goal-driven
Model granularity	Process-centric	Activity-centric
Modality	What must	What must, ought and can
Assumption bias	Over-specified	Under-specified
Alteration	Design time	Design and run time
Coordination /Worker	Human – Machine	Agent
Coordination/Activity	Coordination \diamond Activity	Coordination = Activity
Activity life cycle	Single event	Multiple life cycle event
Language	Procedural	Declarative

From the result of this comparison, we released that any single approach is not capable of representing all business constructs. This leads to the combination of a rule and a process modeling language covers maximum representational constructs proposed.

3.2. Combining BPMN and SBVR for Modelling Business Processes

On the basis of analysis and comparison business rule languages in Table 1, we have found that integration between business rule and business process model in particularly BPMN and SBVR cover almost the business construct. This integration facilitates business modellers to use uniform methodology and representation for enterprise modelling. Furthermore, this integration also enables identification of business rules affecting process models.

The process of defining business vocabularies and business rules is not easy. It is difficult to formulate consistent and complete sets of concepts and propositions governing business. Furthermore, SBVR specifications are of declarative nature and they define business constraints – constraints on structure and on activities, but not on control flow of these activities, i.e. business processes. Though it is possible to predefine sequences of activities by declarative constraints or even extend SBVR for specifying business process, such a practice is not recommended by business rule methodologists. In contrast, they propose the “separation of concerns” – keeping models of business processes and specifications of business rules separately and not intertwining them, because they are changing independently. Also visual modelling is better suited for definition of business processes. Additionally, modelling of business processes helps define right and consistent business rules as well.

4. A Case Study

To demonstrate the potential of the extension BPMN business process model with SBVR business vocabularies and rules, we use VeTIS, an SBVR-compliant plug-in for the CASE tool MagicDraw UML also it can be used as a standalone tool. VeTIS tool is used for the definition of Business Vocabularies and Business Rules using controlled natural language and one of the important features of SBVR is transformation SBVR's specifications into UML class diagrams with OCL constraints. One can launch the MagicDraw UML tool, launch VeTIS editor from MagicDraw UML menu and specify terms and fact types in Business Vocabulary. Having all necessary terms and fact types you will be able to define business rules in Business Rules Vocabulary and transform the overall specification into UML class diagram with OCL constraints. Magic Draw UML fully supports BPMN-based business process modelling therefore SBVR and BPMN activities are fluently integrated under the same working environment. Let's take a look at how all this integrates into the software development process in Figure 1.

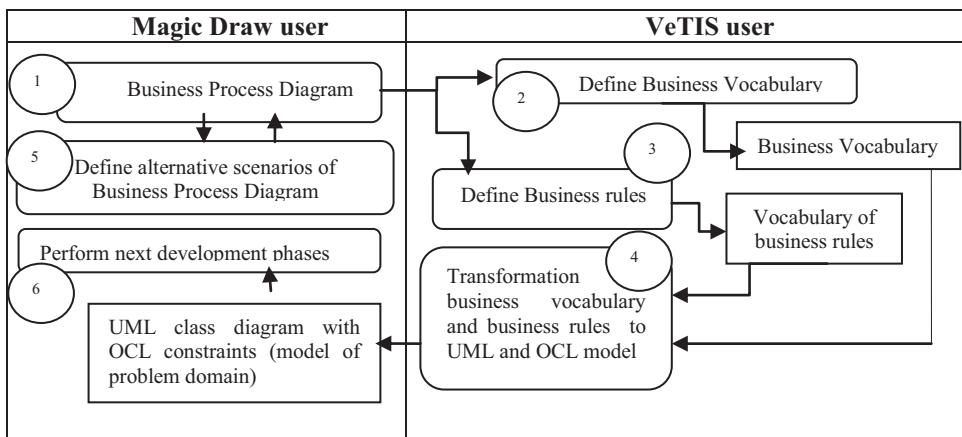


Figure 1. Development process using SBVR business vocabularies and business rules

In order to simulate a step-wise process, we use the simplified example of Bank Information System. In the first step, we choose the loan process and draw this Business Process Diagram using BPMN in Magic Draw environment. In the second and third step, we define business vocabulary and business rules using VeTIS tool. Then we try to transform the vocabulary and rules to UML and OCL model although the model is not complete. In the fifth step, we define alternative scenarios we consider how the process will look in the case of requested loan is not valid.

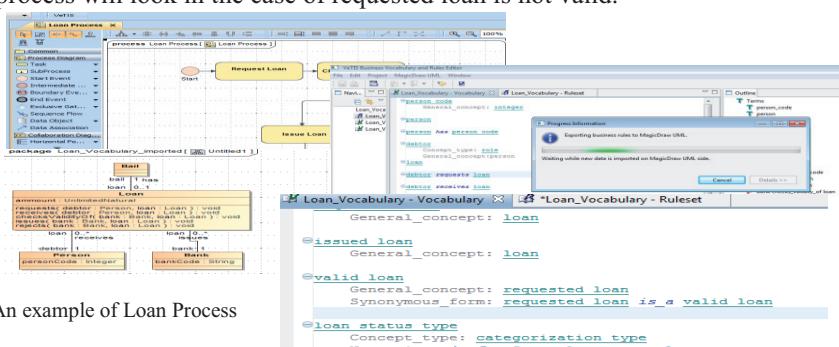


Figure 2. An example of Loan Process

5. Related Work

Early work on the integration of business rules and business processes started appearing shortly after the introduction of the rule modelling concept. Krogstie et al. [11] were the first to suggest that business process and rule modelling approaches should be merged to improve the capturing of temporal information for information systems development. They presented a top-down approach for model specification that involves the use of the External Rule Language (ERL) for specification of process logic at the lowest level of decomposition. This concept was further enhanced by McBrien and Seltveit [18], who presented a way to define the structure of rules within the process model. Knolmayer et al. [19] introduced a framework where process modeling is refined and linked to workflow execution through some layers of Reaction Business Rules. Kappel et al. [9] use Reaction Business Rules to model the coordination in workflow systems. Kovacic [12] developed a metamodel that represents important business constructs (goal, process, activity and events) and technical constructs (data objects, software components, actions in Information Systems). He demonstrates how rules can link these two categories of constructs. Charfi [20] argues that business rules are often hard-coded into web services and proposes a hybrid approach of separating business processes and business rules. Meng et al. [21] introduced a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes. The system uses an event and rule server to trigger business rules during the enactment of workflow processes in order to enforce business constraints and policies at run-time.

Vanthienen and Goedertier proposed an approach to implement SBVR business rules into business processes management life cycle using a service oriented architecture (SOA) [13]. They presented the architecture, which consisted of three layers: business rules and business process layer, services and components layer, application layer. Ali et al. [22] described business rules as separate model used as an integral component of BP modelling. Milanović et al. [8] offered to integrate BPMN with R2ML. They developed a new modelling language rBPMN (Rule-based Process Modeling Language). The main idea of this proposition was to extend some existing elements of BPMN with the Business Rule property. Zhao et al. [23] analysed semantic programming language (SPL) to facilitate the orchestration of semantic web services (SWS). They offered a method to integrate BR and business processes using SWS.

6. Conclusion

In this paper, we have done an analysis and evaluation on business process and rule modelling languages. On the basis of the comparison between them, we have found that business rule or business process approach is not capable of representing all business constructs. Concretely, we have found that integration between business rule SBVR and business process BPMN covers most the business construct. From the result of this, the combination of these two approaches is proposed. BPMN and SBVR are better suited for modelling as they cover maximum constructs. Due to the declarative nature of SBVR, the integration of BPMN and SBVR, features more flexibility and adaptability, is critical requirements for knowledge intensive and dynamic process models. Furthermore, the integration between rule model and process model, which

covers maximum representation constructs using uniform methodology and semantic formulation developed for BPMN will be much more powerful.

References

- [1] OMG, Business Process Modeling and Notation Version 2.0, (2011).
- [2] OMG, Semantics of Business Vocabulary and Business Rules (SBVR) specification, (2006).
- [3] M. Muehlen, M. Indulska, G. Kamp, Business Process and Business Rule Modeling: A Representational Analysis, *EDOC Conference Workshop, 2007. EDOC '07. Eleventh International IEEE* (2007), 189 - 196.
- [4] S. Goedertier, R. Heason, J. Vanthienen, M-BrA2Ce v0.1: A Vocabulary and Execution Model for Declarative Business Process Modeling, (2007).
- [5] OMG, Object Constraint Language Specification Version 2.0, (2006).
- [6] W3C, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, (2004).
- [7] G. Wagner, S. Lukichev, A. Giurca, A. Paschke, J. Dietrich, Verification and Validation of R2ML Rule Bases, *REWERSE IST 506779 Report II-D7* (2006).
- [8] M. Milanovic, D. Gasevic, G. Wagner, Combining Rules and Activities for Modeling Service-based Business Processes, *12th Enterprise Distributed Object Computing Conf* (2008), 11–22.
- [9] G. Kappel, W. R. S. Rausch-Schott, Coordination in workflow management systems - A rule-based approach, *Coordination Technology for Collaborative Applications: Organizations, Processes and Agents* (2007), 99-119.
- [10] H. M. et al, Business process modeling languages: Sorting through the alphabet soup, *ACM Computing Surveys* (2010), 1-56.
- [11] J. Krogstie, P. McBrien, R. Owens, A. H. Seltveit, Information Systems Development Using a Combination of Process and Rule-Based Approaches, *Third Nordic Conference on Advanced Information Systems Engineering* (1991).
- [12] A. Kovacic, Business renovation: business rules (still) the missing link, *Business Process Management Journal* (2004).
- [13] J. Vanthienen, S. Goedertier, How Business Rules Define Business Processes, *Business Rules Journal of Systems and Software* (2007).
- [14] S. W. Sadiq , M. E. Orlowska , W. Sadiq, Specification and validation of process constraints for flexible workflows, *Information Systems* (2005), 349-378
- [15] M. Wesker, W. M. P. van der Aalst, Case handling: a new paradigm for business process support, *Data & Knowledge Engineering* (2005), 129-162.
- [16] P. Green, M. Rosemann, Applying Ontologies to Business and Systems Modelling Techniques and Perspectives – Lessons Learned, *Journal of Database Management* (2004), 105-117.
- [17] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, *Journal of Information Systems* (1993), 217-237.
- [18] P. McBrien, A. H. Seltveit, Coupling Process Models and Business Rules, *Proceedings of the IFIP WG8.1 Working Conference: Information System Development for Decentralised Organisations*, (1995).
- [19] G. Knolmayer, R. Endl, M. Pfahrer, Modeling Processes and Workflows by Business Rules, *Business Process Management, Models, Techniques, and Empirical Studies* (2000), 16-29.
- [20] A. Charfi, M. Mezini, Hybrid web service composition: business processes meet business rules, *2nd international conference on Service oriented computing* (2004).
- [21] J. Meng, S. Y. W. Su, H. Lam, A. Helal, Achieving dynamic inter-organizational workflow management by integrating business processes, events and rules, *35th Hawaii International Conference on System Sciences* (2002).
- [22] S. Ali, T. Torabi, and S. Ben, Rule component specification for Business Process Deployment Database and Expert Systems Applications, *18th International Workshop on Database and Expert Systems Applications* (2007), 595–599.
- [23] S. Y. K. Zhao, L. Zhang, L. Hu, Achieving Business Process and Business Rules Integration Using SPL Future Information Technology and Management Engineering (FITME), *Future Information Technology and Management Engineering (FITME) International Conference* (2010), 329–332.

An Ontological Approach for Collaborative Business Processes formulation with Consensus Methodology

Trung Van NGUYEN^a, Minh T. N. HOANG^a and Hanh Huu HOANG^{b,1}

^a*Faculty of Information Technology*

Hue University

77 Nguyen Hue Street, Vietnam

^b*Department of International Cooperation*

Hue University

3 Le Loi Street, Vietnam

Abstract. Dynamic cross-enterprise collaboration is one of challenges on the business-to-business integration (B2Bi) research nowadays. Semantic Web-based approaches for BPM have been foreseen as a promising solution with taking advantages of Semantic Web technologies such as ontologies, Semantic Web services. With the support of Semantic Web technologies, the gap between business and information technology (IT) communities has been reduced in order to tackle the challenge. Together with the development of web services and its extension with Semantic Web augmented one—Semantic Web Services (SWS), the recent approaches in B2Bi focus on their business process integration between enterprises using SWS. Taking into account the challenge of dynamically forming collaborative business processes with attached services into the execution, in this paper, we proposed an ontological approach for forming a dynamic collaboration of business processes within our BizKB systems for across-enterprise collaboration using the consensus theory in knowledge processing. We focus on an dynamic service discovery based on the consensus methodology.

Keywords. BPM, Semantic BPM, Ontology, Semantic Web, virtual enterprise, Web Services, Consensus methodology.

1. Introduction

Business-to-business integration (B2Bi) or so-called cross-enterprise collaboration in some contexts is one of priority strategies of the e-business research to improve enterprise excellences [1]. It requires exchanging and share business processes between business partners such as customers, suppliers, and distributors. One of the most important challenges in integrating or collaborating between companies in the e-business environment is how to collaborate business processes automatically, accurately, flexibly and effectively. The success of the integration between businesses requires forming and managing collaborative business processes to achieve business

¹ Corresponding Author: Hanh Huu HOANG, Department of International Cooperation, Hue University, 3 Le Loi Street, Hue, Vietnam. Tel: +84-54-3894555. Email: hhhanh@hueuni.edu.vn

goals. Therefore, Business Process Management (BPM) is interested by scientists and the business managers.

Semantic business process management (SBPM) emerges as a promising solution to bridge the gap between businesses and information technology field with the approach to perform business actions which are supported by the information technology with perspective of business process rather than technical perspective [2]. Managing businesses processes shall include methods, techniques and tools to support in designing and constructing rules, managing and analysing businesses operations. However, handling of the BPM automatically in integrating business processes among enterprises is still low due to the interaction between the business process collaboration's semantics. To solve this problem, many researchers have recently proposed solutions in apply artificial intelligence in managing the processes of the collaboration between enterprises discussed in [3].

In our previous work [4], we proposed an approach called Ontological HTN (O-HTN) based on HTN Planning [5] and Web Service Modeling Ontology (WSMO) for forming collaborative business processes dynamically for the cross-enterprise collaboration. The research results the CBP formed with help of O-HTN and attached services profiles. The next part is to discover the appropriate web services to match with service profiles kept in ontologies. In this paper, we introduce an approach using consensus methodology which is originated from solving conflict of data versions [6].

With these motivations, the paper is structured as follows: BizKB Framework [3] is briefly described in the following section. Section 3 introduces background method of HTN planning supported by WSMO; and we identify phases for the business collaboration in Section 4. In Section 5, we use an ontological HTN planning for forming collaborative business processes with an automatic task decomposition solution attached by web services. The paper is concluded with a sketch of future work.

2. Background

2.1. BizKB Architecture

The ultimate goal of our BizKB approach is to build a platform for BP discovery and integration based-on Semantic Web technologies, which supports the process of cross-enterprise collaboration. Many initiatives restrict the range of standards they deal with for political, practical or technical reasons. For companies exposed to different national, industry or enterprise-specific standards – as is practically every business if all of its communications are addressed – this approach is clearly of low practical value. A universally usable methodology will avoid the predefinition of a range of manageable standards [7].

2.1.1. BizKB

As depicted in **Figure 1**, the overall conceptual architecture of the BizKB framework consists of two main parts: the BizKB and the Process Formulator. The output of BizKB framework is CBP with Semantic Web Services profiles attached to the CBP. BizKB is the heart of the BizKB Framework which contains the knowledge of the businesses in the form of Business Process Modeling Ontology (BPMO)-based collaborative business processes with different levels of the abstraction [3].

In order to formulate these BPMO-based processes to store in the BizKB, the BP analysts are required as an important human factor of the system. Based on the analysis on the BPs, the found CBP patterns, level of the abstraction and associate business rules are also extracted and realised.

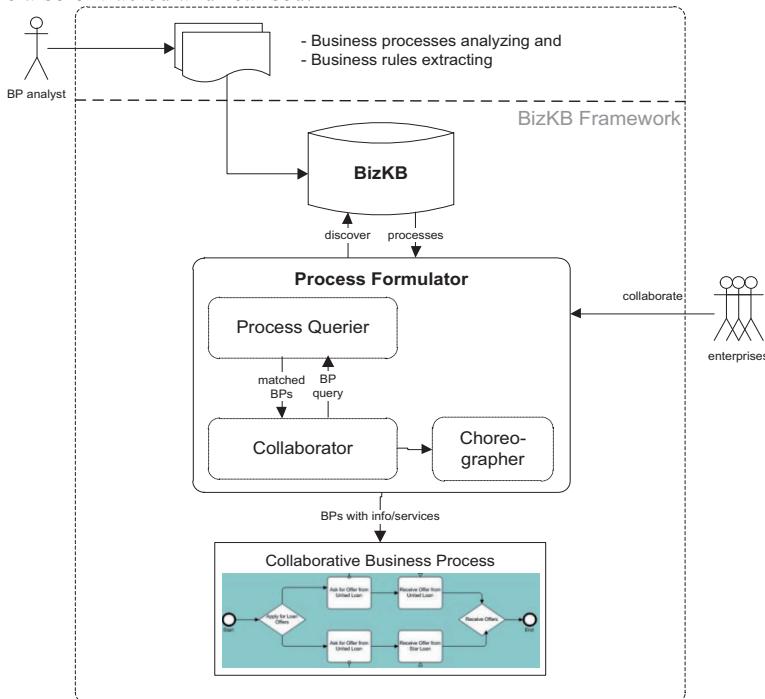


Figure 1. BizKB conceptual architecture

As described in Figure 1, extracted artifacts of BPs are modelled using BPMO according to specific domains and kept in the BizKB. This repository is considered as the process feeder for the later stage of the CBP pattern discovery and CBPs formulation.

Establishing a complete reference collection as a knowledge base beforehand is very unlikely due to the number of standards, their evolution speed and the cost a complete analysis would create, if it were at all possible. Thus the knowledge base has to be flexible, in the sense that its evolutionary growth is not only possible but also a substantial building criterion. Clearly, an approach that does not start with a fully developed knowledge base shows weaknesses in the starting phase. Due to its initially small knowledge base, references supplied by the system might be erroneous and incomplete. But with the growth of the knowledge base, quality improvement occurs quickly [7].

2.1.2. BizKB Ontology for CBP

From above three B2B collaboration phases, a comprehensive list of CBP tasks can be modelled in BizKB Ontology (BO). First, the sequences and hierarchies of granular tasks were synthesised into the three B2B collaboration phases.

BO is a set of ordered compound or primitive task and methods. Compound tasks have one more “*hasMethod*” property since they can be decomposed into primitive tasks that can be performed directly using O-HTN. Each method has a prescription for how to decompose some task into a set of subtasks, with different restrictions that must be satisfied in order for method to be applicable and also various constraints of the subtask and relationship among them.

2.2. Consensus Methodology

Consensus methods were known in ancient Greece and were applied mainly in determining vote results. Along with the development of software methods we can see consensus methods can be applied into many applications fields, especially in solving conflicts and reconciling inconsistent data [6]. In this section, we would like to summarise the core part of the consensus methodology is the algorithms for processing inconsistency of knowledge. Consensus methods are used in a process of solving data inconsistency can be described as follows:

1. Defining the set of potential versions of data;
2. Defining the distance function between versions;
3. Selecting a consensus choice function;
4. Working out an algorithm for consensus choice.

The most important step and it makes the different is the second one, talking of the distance function between different versions of data. Applying this approach into a dynamic mapping method for semantic web services composition in different domains is the key of our approach.

3. Collaborative Service Discovery for BizKB Framework

We have proposed an approach, O-HTN for dynamic collaborative B2B using Web Service Modeling Ontology (WSMO) as the modelling foundation [4]. The main reason for the creation of O-HTN-based BizKB is: O-HTN approach is feasible for dynamically creating CBP task sequences which is significant for a high level tasks composition for formed CBP. Next is about finding the appropriate web services to fill in the service profiles and this will be done by an ontology matching process.

In a collaborative environment, services are composed from different domains and contexts. Therefore, we can be challenged by the inconsistency of different knowledge-based domains. In our approach, we use the consensus methodology [6] to solve this problem.

3.1. The consensus-based semantic distance calculation

An ontology-matching algorithm is based mainly on the semantic measure.

Based on distance functions in consensus methodology [6] together with semantic distance calculation on hierarchical structure of ontologies, we propose a semantic distance algorithm as follows:

Input: Classes C_1, C_2, C_0

Output: Semantic distance between C_1 and C_2

Algorithm:

```

Semantic_Distance(C1,C2)
{
If C1, C2 are similar
    Semantic_Distance(C1,C2) = 0      (1)
Else If exist the direct connection between C1 and C2
    Semantic_Distance(C1,C2) = w * d(C1,C2)  (2)
Else If exist the indirect connection between C1 and C2
    
$$\frac{\sum_{C_0 \in SPath(C_1, C_2)} (w_{C_0} * d(C_1, C_2))^i}{k(k+1)}$$
          (3)
Else
    Semantic_Distance(C1,C2) = 
$$\min \left( \frac{\sum_{C_0 \in SPath(C_1, C_2)} (w_{C_0} * d(C_1, C_2))^i}{k(k+1)} \right)$$
 (4)
}

```

where:

- C₁, C₂: two classes (concepts) in ontology
- C₀: the indirect connection in the cases of (3) or (4).
- w : the weight between these two classes in case of existing direct connection.
- w_{C₀} : the weight between these two classes in case of existing indirect connection - (3) or (4).

The meaningful of the idea is that instead of considering the distance of weight value and conflict concept, we determine smallest distance between experts' opinion reflected in classes of ontologies about services profiles. In this case, we calculation the distance between service requests and services profiles to find out the appropriate services for our needs, and skip the process of consider all services.

In the case of the proposed solution for conflict of service requests is independent to solutions of each service requesters, and the consensus methodology is the right choice. However, the consensus solution must reflect the all solutions and a consensus accepted by all service requestors.

3.2. Consensus-based Collaborative Service Discovery

The ontology matching process is the results of alignments made by the semantic distance calculation. In order to discover the appropriate web services, we need to have an algorithm for evaluation the degree of semantic matching between classes. We call our approach is "collaborative service discovery" because of searching services from different partners with respect to their contexts and domain, and finding out a "consensus" solution for appropriate services.

3.2.1. Algorithm for semantic matching degree of outputs

Input: Outputs for requested and advertised services

Algorithm:

degreeOfmatch(req.out, adv.out)

{

```

If (req.out = adv.out) or
    (subclassOf(req.out,adv.out)==true) then
        Semantic_Distance(req.out,adv.out)=0
    Else
        If subsumesOf(adv.out,req.out)==true then
            Semantic_Distance(adv.out,req.out)=w*d(adv.out,req.out)
        Else
            If subsumesOf(req.out,adv.out)==true then
                Semantic_Distance(adv.out,req.out)=
                    
$$\frac{\sum_{C_0 \in SPath(C_1,C_2)} (w_{C_0} * d(req.out,adv.out))^i}{k(k+1)}$$

            Else
                Semantic_Distance(adv.out,req.out)=
                    
$$\min \left( \frac{\sum_{C_0 \in SPath(C_1,C_2)} (w_{C_0} * d(req.out,adv.out))^i}{k(k+1)} \right)$$

    }
}

```

3.2.2. Algorithm for semantic matching degree of input

Input: Inputs for requested and advertised services

Algorithm:

```

degreeOfmatch(req.inp, adv.inp)
{
    If (req.inp = adv.inp) or
    (subclassOf(req.inp,adv.inp)==true) then
        Semantic_Distance(req.inp,adv.inp)=0
    Else
        If subsumesOf(adv.inp,req.inp)==true then
            Semantic_Distance(adv.inp,req.inp)=w*d(adv.inp,req.inp)
        Else
            If subsumesOf(req.out,adv.out)==true then
                Semantic_Distance(adv.inp,req.inp)=
                    
$$\frac{\sum_{C_0 \in SPath(C_1,C_2)} (w_{C_0} * d(req.inp,adv.inp))^i}{k(k+1)}$$

            Else
                Semantic_Distance(adv.inp,req.inp)=
                    
$$\min \left( \frac{\sum_{C_0 \in SPath(C_1,C_2)} (w_{C_0} * d(req.inp,adv.inp))^i}{k(k+1)} \right)$$

    }
}

```

where:

+ *Semantic_Distance()* is the semantic distance calculated by the proposed algorithm.

$+ d(\text{adv.in}, \text{req.inp})$ or $d(\text{adv.out}, \text{req.out})$ are the matching degree between services. The semantic matching degree is dependent to the ontology structure and RDFNode.

Similarly, these algorithms can be applied into requested services in service profiles and advertised services from different contexts. The solutions can be achieved with the basis of matching degree, multi-level matching and the combination in different levels of range of the function *degreeOfmatch()*.

4. Ontological Consensus Approach in BizKB Framework

We have proposed O-HTN for dynamic collaborative B2B using Web Service Modeling Ontology (WSMO) as the modelling foundation [4], WSMO is a flexible ontology language with dynamic reasoning features, supports execution based-on Web services as well. BO describes the hierarchical relationships between compound and primitive B2B collaboration tasks, and methods for task decompositions, and relevant planning criteria (e.g. cost, quantity ordered, type of collaboration) embedded in the methods. Different criteria input by the user result in different permutations of sub-tasks.

4.1. O-HTN based Process Formulator

The interactive part of the BizKB framework [3] is the *Process Formulator* component which consists of two main subparts – *Process Querier* and the *Collaborator*. These parts are interacted by the demanding enterprise to find out the appropriate CBP patterns to form a collaborative business process with the help of the third subpart - *Choreographer*.

The Process Querier helps to find the appropriate process patterns at a certain abstraction level. Due to the enterprise's discovery into the BizKB, the detailed level will be matched to the need. For example, in the Order Management process, one participant wants to identify the process of "Buy" products, however the participant cannot clearly identify parts of the process and related information, the Process Querier can help to identify the basic patterns, sample processes, and even the generalization levels of the needed process. After matched processes returned, the Choreographer will coordinate to finalize the output collaborative business process to fulfil the B2B integration demand. Here, we use O-HTN Algorithm as described in following subsection for this phase.

The new-formed CBP is attached with WSMO services profiles for specific Semantic Web Services. This process is serialized using WSMO standard which conforms the unification of the framework's BPMO standard (which is based on WSMO) and benefits from Semantic Web Services advantages.

4.2. Process Formulation Workflow

The O-HTN based Architecture for the Process Formulator is described in **Figure 2**. User' request is presented in WSMO ontologies and a WSMO Goal.

In the next step, WSMX uses the discovery component to find web services profiles which have semantic descriptions registered through their capabilities and interfaces. A set of properties strictly belonging to a goal is defined as non-functional properties of a WSMO goal. A goal may be defined by reusing one or several already-existing goals by means of goal mediators.

During the discovery process the users' Goal and the web services description may use different ontologies. If this occurs *Data Mediation* is needed to resolve heterogeneity issues. Once these mappings are registered with WSMX, the runtime data Mediation component can perform automatic mediation between the two ontologies. Once this mediation occurred and a given service that can fulfil the user's goal is chosen, WSMX can begin the process of invoking the service.

If there is no single web service that satisfies the request then the request will be offered to the planner. The planner then tries to combine existing Semantic Web services and generate the process model. In the proposed framework, the process generator is based on HTN-planning. The process generator to tackle the problems of heterogeneous ontologies and choreography uses discovery component of WSMX. Thus via this component, the process generator will be able to discover the appropriate Semantic Web services for dynamic cross-enterprise collaboration. Finally the process model will be offered to the WSMX for its execution. The stages for execution of Web services as a process model are like as single web services.

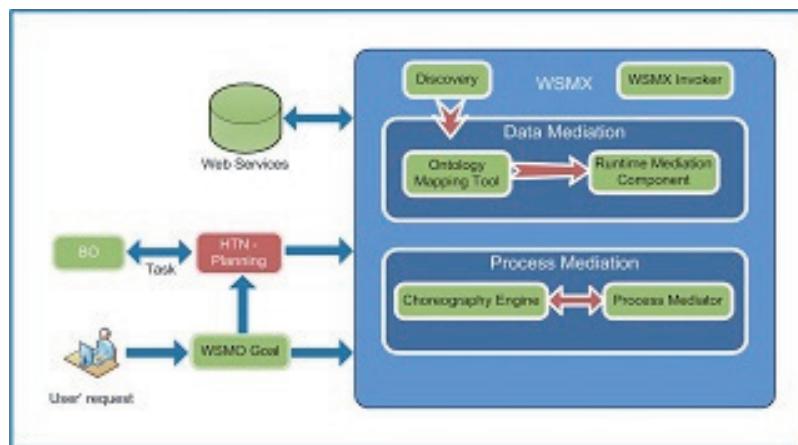


Figure 2. The O-HTN-based Process Formulator architecture

4.3. Collaborative BizKB with O-HTN and Consensus Service discovery component

Applying these algorithms together with O-HTN in our approach, we structure the new process for CBP formulation and service discovery for BizKB framework as follows:

Collaborative business processes are dynamically formed by using O-HTN algorithm in a flexible task decomposition process. Formed CBP with service profiles will then hand-in the services discovery phase for finding appropriate services.

The consensus service discovery based on newly invented ontology matching process helps to find out requested services in a better solution in collaborative service partner providers. The discovered services will be then transferred into the choreography mechanism for forming executable business processes.

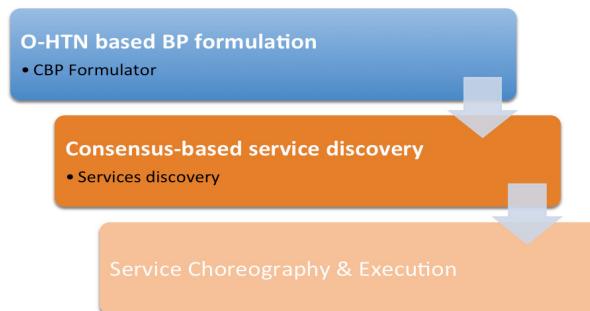


Figure 3: A collaborative Biz-KB process with support of O-HTN and Consensus service discovery mechanisms

5. Related Work

Since the failure of the non-semantic approaches as mentioned above, research efforts have been emerged from the motivation of knowledge management and applying Semantic Web technologies into BPM researches to bring the administrative side and IT side together.

SUPER [8] addresses the ever enduring need of new weaponry in struggle for survival in optimistic business environment where profit margins dramatically drop while competitiveness reaches the new sky high limits. The major objective of the SUPER project is to raise BPM to the business level, where it belongs, from the IT level where it mostly resides now [8]. This objective requires that BPM is accessible at the level of semantics of business experts. SUPER's approach has tried to transform existing BPMN and BPEL standards into a semantics-enriched form, respectively called sBPMN and sBPEL [9, 10] in the attempt to realize their goals.

Haller in [11] extended the *multi metamodel process ontology* (*m3po*) introduced with concepts for a full formalisation of the meta-model of XPDL. In the context of their approach, to deal with collaborative processes (choreographies) these internal workflow models are aligned to the external behaviour advertised through web services interfaces. The *m3po* ontology presented explicitly models the complete semantics of XPDL. The integrated *m3po* is used as shared representation to perform the integration. The advantage of this approach is that authors use a web ontology language to formalise proposed model into linked data with established business document standards.

One of recent efforts in cross-enterprise collaboration research is Genesis approach based on its ontology called Business-OWL (BOWL) [12]. The core of the approach is about BOWL that is a hierarchical task networking (HTN) modelled in OWL describing the hierarchical relations between tasks of collaborative business processes consist of compound tasks, primitive tasks and task decomposition methods. for a suitable solution for current dynamic e-commerce today. Therefore, the knowledge described by HTN needs to be modelled in forms of OWL ontologies proposed in this approach. Following this research direction brings new opportunities, new prospects and useful tools for e-business and B2B integration especially. The effort follows this line is Jung's work [2] which focus on basic problems of applying ontology aligning for business process integration.

6. Conclusion and Outlook

In this paper we have proposed a collaborative services discovery mechanism with the support of the consensus methodology. Together with ontology-based approach using Ontological-HTN and WSMO for forming collaborative business processes in the dynamic cross-enterprise collaboration [4]. The approach is motivated by the semantic web approach in efforts of bridging business perspective and IT world together, and provides an architecture that supports the dynamic semantics-based collaborative business process management in a new e-business environment. We have successfully implemented O-HTN and the consensus-based ontology matching algorithms for web services discovery with some improvements in compared to [13, 14] which have not discussed the service discovery issue for CBP.

For the future work, we plan to improve the algorithms with case studies and do some experiments with mapping of attached web services into the execution level with practical examples.

References

- [1] H. Hoang, P.-C. Tran, and T. Le, State of the Art of Semantic Business Process Management: An Investigation on Approaches for Business-to-Business Integration, presented at the Intelligent Information and Database Systems, 2010.
- [2] J. J. Jung, Semantic business process integration based on ontology alignment, *Expert Systems with Applications*, vol. 36, pp. 11013-11020, 2009.
- [3] H. Hoang and T. Le, BizKB: A Conceptual Framework for Dynamic Cross-Enterprise Collaboration, presented at the Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, 2009.
- [4] V. M. Hoang and H. H. Hoang, An Ontological Approach for Dynamic Cross-Enterprise Collaboration, presented at the WAINA-2012, Fukuoka, Japan, 2012.
- [5] K. Erol, J. Hendler, and D. S. Nau, HTN planning: Complexity and expressivity, presented at the Proceedings of the National Conference on Artificial Intelligence, 1995.
- [6] N. T. Nguyen, Using Consensus Methodology in Processing Inconsistency of Knowledge Advances in Web Intelligence and Data Mining. vol. 23, M. Last, P. Szczepaniak, Z. Volkovich, and A. Kandel, Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 161-170.
- [7] C. Pedrinaci, J. Domingue, C. Brelage, T. van Lessen, D. Karastoyanova, and F. Leymann, Semantic Business Process Management: Scaling Up the Management of Business Processes, presented at the 2008 IEEE International Conference on Semantic Computing, 2008.
- [8] M. Born, C. Drumm, I. Markovic, and I. Weber, SUPER - Raising Business Process Management Back to the Business Level, *ERCIM News*, vol. 70, pp. 43 - 44, 2007.
- [9] M. Dimitrov, A. Simov, S. Stein, and M. Konstantinov, A BPMO Based Semantic Business Process Modelling Environment, presented at the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), Innsbruck, Austria, 2007.
- [10] Z. Yan, E. Cimpian, M. Zaremba, and M. A. M. M. Mazzara, BPMO: Semantic Business Process Modeling and WSMO Extension, presented at the Web Services, 2007. ICWS 2007. IEEE International Conference on, 2007.
- [11] A. Haller, M. Marmolowski, W. Gaaloul, E. Oren, B. Sapkota, and M. Hauswirth, From Workflow Models to Executable Web Service Interfaces, presented at the Proceedings of the 2009 IEEE International Conference on Web Services, Los Angeles, CA, USA, 2009.
- [12] R. Ko, A. Jusuf, and S. Lee, Genesis Dynamic Collaborative Business Process Formulation Based on Business Goals and Criteria, presented at the World Conference on Web Services - I, 2009.
- [13] A. J. Ryan K. L. Ko, S. G. Lee, Genesis - Dynamic collaborative business process formulation based on business goals and criteria, 2009.
- [14] S. G. L. Ryan K.L.Ko, E.W.Lee, Andre Jusuf, Dynamic collaborative business process formulation via ontologised Hierarchical task network (htn) planning, 2009.

Subject Index

abstraction	91	data grid systems	169
ad hoc supply chain	434	data mining	356
adjustable autonomy	335	deduction	203
agent systems	3	design patterns	203
agent theory	293	distributed computing	58
agents	30, 40, 203, 404	distributed query processing	169
agent-based computing	181	distribution	276
agent-based model (ABM)	102	dynamic autonomy distribution	335
agent-based modeling	71	EHR	375
agent-based optimization	159	emergence	50
architecture	404	engagement	404
attribute reduction	417	enterprise alliance formation	434
attribute selection	427	enterprise network	434
attribute-value	427	epistemic logics	225
automatic reuse	315	Erlang technology	21
autonomy analysis	335	four-valued logic	255
BDI architecture	395	GAMA	91
BDI reasoning	137	Gaussian process entropy	102
belief bases	3	geostatistics	102
BPEL	203	global state caching	235
BPM	276, 444, 454	global-optimisation	181
bug ontology	213	grammar logics	225
bug search	213	graph	58
bug tracking system	213	graph structure	40
business process	276	health information systems	375
business rule languages	444	high utility itemset	356
business rules	444	HL7 CDA	375
business-to-business market	303	incentive contracts	293
cloud computing	284	incomplete decision table	417
cloud supply chain	284	indeterminate valid time	325
cognitive agent design	395	information extraction	364
collective problem solving	159	intelligent control	40
coloured petri nets	245	intelligent software agents	115
communication	146, 255	intelligent traffic control	137
complex service systems	71	interaction pattern	315
composition	385	interaction protocol	315
conflict resolution	255	interaction server	315
conjecture	146	interlinking	427
consensus method	325	interoperability	375
consensus methodology	454	JADE	276
control loop	276	K-depth	346
cooperation	30	keyword search	346
CPDL	235	knowledge integration	325

Kriging standard error	102	RDF	346
layered adjustable autonomy (LAA)	335	recommendation systems	427
learning	125	reduct	417
least cost	30	regular grammar logics	235
lighting control	40, 58	resource allocation	169
linked open data (LOD)	193, 427	risk aversion	293
linked open data federation	193	robust message	146
load balancing	169	rough set theory	417
logistics outsourcing	293	rule languages	225
management and engineering (SSME)	71	rule-based query languages	3
marketing	267	S5-knowledge model	146
memetic-computing	181	scheduling	169
meta-heuristics	181	selective advertisement	267
meta-model	91	semantic annotation	364
military	404	semantic BPM	454
model-based testing	245	semantic tableaux	203
modeling	81, 276	semantic web	193, 364, 454
multi-agent	276	service science	71
multi-agent coordination	137	service-dominant logic	71
multi-agent platform	21	service-oriented architecture (SOA)	434
multi-agent system	50, 58, 303, 315, 375	shortest path	30
multi-level agent-based modelling	91	simulation platform	395
multimodal services	81	simulation	81, 91, 137, 276, 404
multi-scale	91	situation awareness	335
Nash equilibrium	146	slashed graphs	58
NKCS Model	71	social networks	267
non-corporative game	146	social simulation	267
normative protocol identification	115	software agent	335
norms detection	115	solving	159
norms mining	115	SPARQL-endpoint	193
online logistics exchanges	284	speech acts	255
ontology	385, 454	sport ontology	364
ontology integration	193	state machines	30
optimal design	102	strongly-typed genetic programming	50
optimization	102, 169	structure index	346
outdoor lighting	40	supply chain	284
paraconsistent modelling	255	surveillance network	102
patient summary	375	synthetic battlefield	404
peer-to-peer network	213	system design	404
personalization services	303	tableaux	235
petri net	385	task allocation	125
place/transition petri nets	30	team formation	125
population-based methods	159	temporal logic	203
privacy preserving	356	tolerance rough set	417
protocol	146	tractable models	255
random	276	transport systems	30
		transportation	81

TTCN-3	245	vehicle routing problems	159
unit disk graph (UDG)	102	verification	203
users' skills and preferences	303	virtual enterprise	454
utility	356	web services	203, 385, 454
value co-creation	71	word-of-mouth	267

Author Index

Ahmad, A.	115, 335	Joo, J.	385
Ahmad, M.S.	115, 335	Jung, J.J.	427, 434
Annamalai, M.	335	Kang, S.	346
Bae, M.	346	Khalfaoui, S.	315
Bagić Babac, M.	245	Kim, J.	404
Barbucha, D.	vii, 159	Kisiel-Dorohinicki, M.	181
Basheer, G.S.	335	Klimek, R.	203
Batista Júnior, A.d.A.	137	Korczyński, W.	181
Byrski, A.	181	Kotulski, L.	40
Cao, T.-D.	364	Kovács, G.	284
Čapkovič, F.	30	Kurahashi, S.	267
Chung, Y.	404	Le, B.	356
Claudi, A.	375	Le, M.N.	102
Coutinho, L.R.	137	Le, M.T.	vii
Dinh, V.V.	417	Le, S.T.	213
Dolcini, G.	375	Le, V.-M.	395
Dragoní, A.F.	375	Lejouad Chaari, W.	315
Drogoul, A.	91, 102	Lénárt, B.	284
Dunin-Kęplicz, B.	255	Lin, C.-W.	356
Duong, T.H.	193	Mahmoud, M.A.	115
Epimakhov, I.	169	Małuszyński, J.	3
Ernst, S.	40	Marakhimov, A.	385
Fung, R.Y.K.	293	Matsuhisa, T.	146
Gaudou, B.	395	Morvan, F.	169
Giang, N.L.	417	Mostafa, S.A.	335
Grzybowska, K.	284	Mustapha, A.	115
Ha, S.V.U.	213	Nguyen, C.D.	193
Hagino, T.	364	Nguyen, D.	346
Hamano, D.	125	Nguyen, H.-C.	364
Hameurlain, A.	169	Nguyen, L.A.	225, 235
Hamid, N.H.A.	115	Nguyen, M.	356
Hayano, M.	125	Nguyen, N.T.	325
Hoang, H.H.	444, 454	Nguyen, Q.-M.	364
Hoang, M.T.N.	454	Nguyen, T.V.	454
Hong, T.-P.	356	Nguyen, V.D.	325
Howlett, R.J.	vii	Oh, S.	346
Huynh, H.X.	102	Othman, A.	81
Huynh, T.K.	213	Palazzo, L.	375
Hwam, W.K.	404	Park, S.C.	404
Jain, L.C.	vii	Pham, X.H.	427
Jevtić, D.	245	Phan, T.-H.	364
Jezic, G.	303	Rajapakse, C.	71

Rossi, M.	375	Tran, H.M.	213
Scemama, G.	81	Trung, D.N.	434
Sędziwy, A.	58	Truong, V.X.	102
Sernani, P.	375	Turek, W.	21
Shimohara, K.	50	Verbrugge, R.	255
Skocir, P.	303	Vo, B.	356
Šperka, R.	276	Võ, Đ.-Â.	91, 395
Spišák, M.	276	Vu, V.V.	356
Strachocka, A.	255	Vymětal, D.	276
Sugawara, T.	125	Wojnicki, I.	40
Szałas, A.	3, 225, 255	Yim, J.	385
Taillandier, P.	395	Yin, S.	169
Takeda, H.	427	Yusoff, M.Z.M.	115
Tanev, I.	50	Zargayouna, M.	81
Terano, T.	71	Zeddini, B.	81
Tian, K.	50	Zhu, Q.	293
Tran, C.P.T.	444	Zucker, J.-D.	91